ECES 680 - Winter 2018 Drexel University
Instructor: Matthew Stamm

Assignment 2

This assignment is due by 11:59 p.m. on Monday, February 5. All reports should be submitted as PDFs.

JPEG is one of the most widely used image compression standards.

- Use the Matlab function *imwrite* to save the files **peppers.tiff** and **baboon.tiff** using quality factors of 90, 70, 50, 30, and 10. Compute the PSNR between each original image and its JPEG compressed version at each quality factor. Additionally, record the file size of the image when it is saved at each quality factor. What is the relationship between the image's file size and its quality? What distortions are introduced by JPEG compression? Why do you think they occur? At what quality factor do these distortions become unacceptably strong?

- Write two Matlab function that implement your own JPEG-like encoder and decoder. The encoder should follow the following process:

  1. Segment the image into 8×8 pixel blocks.
  2. Compute the DCT of each block.
  3. Quantize these DCT coefficients using a user specified quantization table Q.
  4. Reorder each block of quantized DCT coefficients into a one-dimensional sequence using zig-zag scanning. You can use *ZigzagMtx2Vector.m* that is provided to you to perform zig-zag scanning and use *Vector2ZigzagMtx.m* for reconstructing the matrix from a zig-zag scanned sequence.
  5. Encode the resulting sequence. For Entropy Encoding, use the *JPEG_entropy_encode.m* module provided. This function will read a matrix, in which each row represents a vectorized DCT block, write a bit stream whose filename is always named as **JPEG.jpg**, and return the length of this file. *JPEG_entropy_encode.m* is an interface for generating a text file, *JPEG_DCTQ_ZZ.txt*, and running *jpeg_entropy_encode.exe*. For the entropy decoding, use *JPEG_entropy_decode.m*, which performs the inverse functionality.

  The decoder should reconstruct the image by performing each of these steps in reverse. Please comment your code and append it to your report.

- Use your JPEG-like encoder to encode the image **peppers.tif**. First do this using the standard JPEG luminance quantization table

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{1}$$

  Record the image's file size and the PSNR between the original and decompressed image. Next, try changing the quantization table and encoding the image. Is it possible to achieve both a lower file size and a higher PSNR? Include the quantization table that you design in your report.