Communication and Information Engineering Program
Information Theory and Coding (CIE 425)
Fall 2022
JPEG Assignment

This is an assignment in which you are required to develop a JPEG encoder and decoder for any image of your choice with 8-bit grey scaled pixels.

Some notes on the Discrete cosine transform (DCT) that you might find useful

1- Please use the DCT basis functions provided in the text book

$$b[x,y] = \cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

2- Divide the DCT coeffs at (u=0,v=0) by 64

3- Divide the DCT coeffs with either u = 0, or v = 0 (not including the (0,0) coeff) by 32

4- Divide the remaining DCT coeffs by 16

5- To perform the IDCT operation, you should just multiply each DCT coefficient by the corresponding basis function, and then add them without any further scaling

The above notes are to ensure that the output after performing DCT and IDCT restores the original pixel values. You should check on a single 8x8 block your DCT and IDCT codes and make sure that this is the case

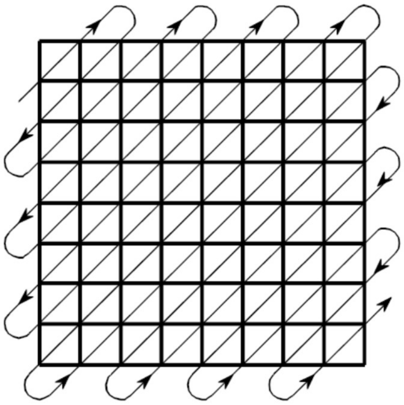
<u>You are required to perform the following steps using your own developed code:</u>

1- Read and divide the image into blocks of 8x8 pixels

2- Perform DCT on each block (develop your own DCT, don't use the ready-made function in MATLAB)

3- Perform the quantization step per 8x8 block using at least two quantization tables

4- Transform each block from 2-D into 1-D vector using the following pattern



5- Use run-length encoding to compress the stream of zeros that may results due to the quantization (use your own developed code)

6- Use the built-in Huffman function in MATLAB (or elsewhere) to encode the final stream into a further compressed bit stream

Decoder starts here

7- Use the built-in Huffman decoder function in MATLAB (or elsewhere) to decode the Huffman encoded stream
8- Perform run-length decoding (use your own developed code)
9- Transform the 1-D vector into groups of 8x8 matrices
10- Multiply each group by the quantization tables
11- Perform IDCT using your own developed function on each 8x8 pixel group
12- Combine the 8x8 pixel groups into a single image and save it back to a file
13- Compare the original image with the compressed image when using each quantization table in step (3)

Deliverables:

1- Define the specifications of your JPEG compression algorithm. The specifications should include, e.g., image quality after compression, compression ratio, etc.
2- Your own developed JPEG encoder and decoder MATLAB (or Python) code with all the previous steps clearly implemented as indicated above. The correctness of the code will be graded. A documentation of your code is required along with the code itself using **either** MATLAB **Live Editor**, or Python **Jupyter Notebook**. In the documentation you should explain clearly all your used modules (functions) and their corresponding inputs, outputs, internal variables, etc., and how they map to the JPEG algorithm. You should also display the relevant outputs of the modules in your live editor or notebook.
   This is a short video about the MATLAB live editor:
   https://www.youtube.com/watch?v=bu4g8ID3aEk
   And another one about Python Jupyter Notebook:
   https://www.youtube.com/watch?v=3C9E2yPBw7s
3- Calculate the compression ratio for the compressed images along with a quantitative measure of the image quality after compression. Comment also qualitatively on the compressed images versus the original image for each quantization table.
4- A justification for the choice of your quantization table from at least two options based on the specifications defined in deliverable (1).
5- The image before and after compression.