Programming assignment no. 3

Implement your own version of jpeg compression in its old version (before jpeg 2000)

The algorithm is as follows:

1. Read an image from a file into a matrix.

2. Divide the image into blocks of $n \times n$ pixels where $n < N$ (typical value $n = 8$).

3. Calculate the discrete cosine transform of each block

$$a_{k,j} = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} \alpha_k \alpha_j f(x,y) \cos \frac{k(x+\frac{1}{2})\pi}{n} \cos \frac{j(y+\frac{1}{2})\pi}{n}, k,j = 0,...,n-1,$$

   where $\alpha_k = \begin{cases} \sqrt{\frac{1}{n}}, \text{ if } k = 0 \\ \sqrt{\frac{2}{n}} \text{ otherwise} \end{cases}, k = 0,...,n-1.$

4. Mask out some of the coefficients i.e., Multiply each block element by element by a matrix M i.e., calculate

$$b_{kj} = a_{kj} m_{kj}, k,j = 0,...,n-1.$$

   (Typical mask matrix is

$$M = [m_{kj}]_{k,j=0,\ldots,n-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.)$$

5. Write the results into a matrix (image) B.

6. Perform the inverse discrete cosine transform on each block of the image B:

$$\bar{f}(x,y) = \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \alpha_k \alpha_j b_{kj} \cos \frac{k(x+\frac{1}{2})\pi}{n} \cos \frac{j(y+\frac{1}{2})\pi}{n}, x,y = 0,...,n-1.$$

7. Store the results into an image (bitmap).