Advanced Image Processing: Assignment 4 (Due Apr 7, 2025)

Note: Please provide detailed comments for code that may be written to solve the following problems. The assignment will be evaluated not just based on the final results but also how you obtained them. Late submissions will be penalized.

Problem 1: JPEG Implementation (20 points)
Implement a toy version of JPEG through the following steps:

1. Transform: Compute an 8x8 discrete cosine transform (DCT) for every non-overlapping block in the input grey scale image.

2. Quantization: Use the following quantization matrix to quantize each DCT coefficient in a given 8x8 block

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

   Note that the quantized index of the DCT coefficient $x(i, j)$ is given by

$$y(i, j) = \left\lfloor \frac{x(i, j)}{Q(i, j)} + 0.5 \right\rfloor$$

   and the reconstruction is given by $\hat{x}(i, j) = y(i, j)Q(i, j)$.

3. Lossless source coding: Use the following table to encode the quantized index corresponding to each DCT coefficient.

Quantized DCT index	Code
0	0
-1,1	10x
-3,-2,2,3	110xx
-7,-6,-5,-4,4,5,6,7	1110xxx
...	...

The output bitstream (or file) is given by the concatenation of the sequence of bits produced for each 8x8 block.

Using the JPEG implementation described above:

1. Compute the size of the output file generated for the cameraman.tif image provided to you. Also compute the mean squared error (MSE) between the original image and reconstructed image. The reconstructed image is obtained by taking the inverse DCT for each block of quantized reconstructions of DCT coefficients. Calculate the compression ratio (defined as the ratio of the input image in bits and size of the output file in bits).

2. Compute the file size and mean squared error for a JPEG implementation where each DCT coefficient in your JPEG implementation is merely rounded to the nearest integer without any quantization. Compute the compression ratio. Comment on how much compression ratio you can achieve with very minimal MSE.

3. Plot a file size vs MSE curve for JPEG by varying the file size and computing the MSE for each file size. At one end, you need to have a file size of 0 and on the other end, you can have the file size obtained in Part (2) above. (Hint: You need to need to use a global scaling of quantization matrix to obtain any target file size.)

Problem 2: YOLO Object Detection (20 Points):

1. Train a YOLO-v7 model on the following dataset (`https://www.kaggle.com/datasets/biancaferreira/african-wildlife`) with 4 classes of wild animals. The labels and coordinates are available in the corresponding txt files. Each row in the annotation file is of the form class-id center-x center-y width height corresponding to an object in the image. Split this dataset into 80:20 train-test ratio.

2. Evaluate the performance of the trained model on the test set. Compute the following metrics: (i) mAP@0.5IoU; (ii) mAP@[0.5:0.95]IoU (take average of mAP at 10 IoU thresholds from 0.5 to 0.95 with a step size of 0.05). Also, provide some qualitative results in your report.

3. Modify the test images synthetically to mimic real-world scenarios, by adding mild Gaussian blur, contrast changes, etc. using PyTorch transforms. Evaluate the trained model performance again on this modified test set and compare the performance with Part (2).

Prepare a report containing the answers to all the problems and submit along with relevant code.