Advanced Image Processing: Assignment 4 (Due Apr 9, 2024)

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 (15 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. Compare the file size with the file size obtained with any default Python/MATLAB function for JPEG compression. Note that for this comparison, you need to control any input parameters to the default functions such that the MSE is the same for the previous case (the one that you implemented).

Problem 2: Comparison of perceptual quality measures (20 points)

In this problem, you will compare PSNR, SSIM and LPIPS (2 versions) with respect to their performance in terms of correlation with human perception.

A. Algorithms: Evaluate the following metrics on the dataset mentioned in Part (b). You should obtain a single number as quality with respect to each of the metrics below for each image.

1. Peak signal to noise ratio in pixel domain

2. Single scale structural similarity index (you can use default code available in Python or MATLAB)

3. Learned perceptual image patch similarity metric (LPIPS): This is a deep learning based measure that is available online (`https://pypi.org/project/lpips/`). You need to compare two versions here, one that uses VGG and another that uses Alexnet.

B. Performance measurement: Download the database available at
`http://ece.iisc.ac.in/~rajivs/courses/aip2016/hw5.rar`
The database comes with the following:

1. distorted images in the "gblur" folder

2. reference images in the "refimgs" folder

3. reference image name for every distorted image in the "gblur" folder in "refnames_blur"

4. human opinion scores in "blur_dmos"

5. indicator of whether the image in the "gblur" folder is an original image in "blur_orgs"

C. Questions

1. Compute the Spearman rank order correlation coefficient between the dmos scores in "blur_dmos" and each metric you considered in A after having removed the scores that correspond to the original images in the "gblur" folder (as mentioned earlier, this information is contained in blur_orgs).

2. Comment on the relative performances of all the indices

Problem 3: YOLO Object Detection (15 Points):

- Train a YOLO v7 Object Detection model on this dataset.
- The dataset has 7 classes. The classes are defined in *data.yaml* file. Each image has a corresponding label file with bounding box annotations. 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. The values are in zero to one normalized coordinate space. Refer YOLOv5 annotation format if any doubts.
- Evaluate the performance of the trained model on the given 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, give the qualitative results in your report.
- Briefly Summarize the differences between YOLO v1 (taught in class) and YOLO v7 in terms of (1) Architecture design, (2) Loss function, (3) Computation complexity. Justify how the changes proposed addresses the drawbacks in YOLO v1 , if any.

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