1.5em 0pt

Assignment 4

E9 246: Advance Image Processing
Indian Institute of Science, Bengaluru

Name:	Alokendu Mazumder
SR Number:	20134
Department:	Electrical Engineering
Program:	PhD
Code Link:	**Click Here**

1 JPEG Compression

For a given image, we implement a toy JPEG compression algorithm to measure it's performance in image compression and reconstruction. We use the given formulae in assignment for compression as well as reconstruction. We use the following quatization matrix:

$$Q = \begin{bmatrix} Q(a,b,c) & Q(b) \\ Q(b) & Q(b) \end{bmatrix} \tag{1}$$

Where, $Q(a,b,c)$ & $Q(b)$ are defines as:

$$Q(a,b,c) = \begin{bmatrix} c & a & b & b \\ a & a & b & b \\ b & b & b & b \\ b & b & b & b \end{bmatrix} \; Q(b) = \begin{bmatrix} b & b & b & b \\ b & b & b & b \\ b & b & b & b \\ b & b & b & b \end{bmatrix} \tag{2}$$

Also, from the given encoding table in assignment, we found the formulae to compute the number of bits for the DCT coefficient as follows:

$$b(i,j) = \begin{cases} 0 & \text{if } y(i,j) = 0 \\ 2\lceil \log_2(|y(i,j)|+1) \rceil + 1 & \text{otherwise} \end{cases} \tag{3}$$

Here, $b(i,j)$ is the length of bits of the $(i,j)^{th}$ DCT coefficient.

1.1 Results of sub question 1

For the first sub question, where the quantized index of the DCT coefficient $x(i,j)$ is given by :

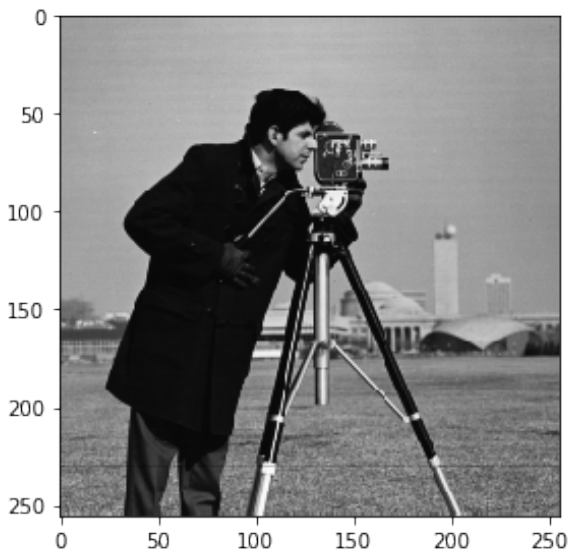
$$y(i,j) = floor\left(\frac{x(i,j)}{Q(i,j)} + 0.5\right) \tag{4}$$

The file size (in bits) of original file is: **981330**

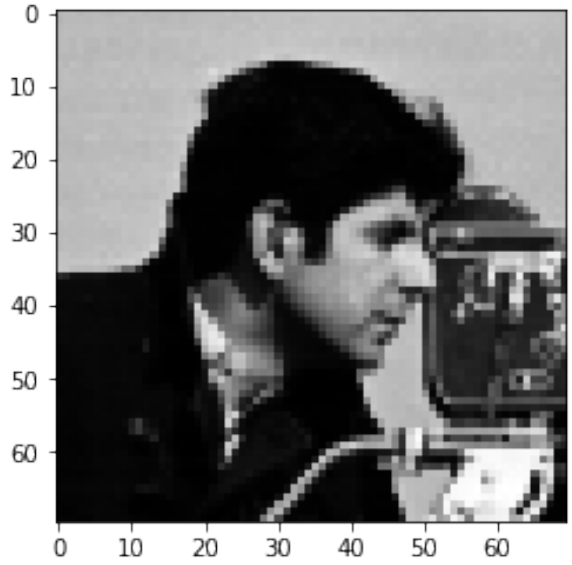
Table 1: Observations for sub-question 1

MSE between original and reconstructed image	Compression Ratio	File Size (in bits)
27.85	9.72	100918

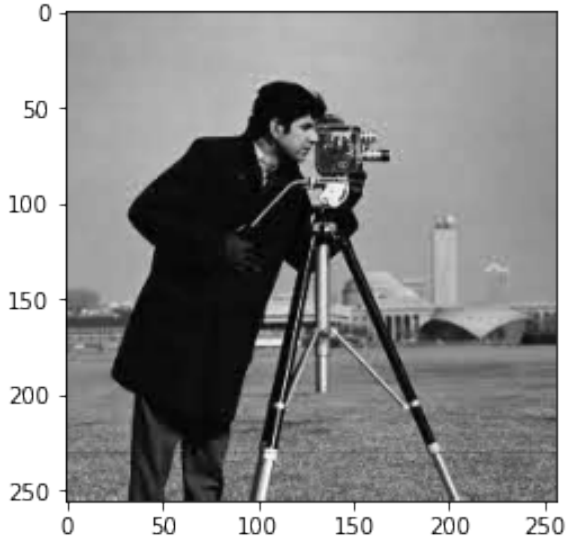
We can see that the implementation provides a good compression of a file about **9.72** times smaller than the original. The MSE on the other hand, is quite large and could result in quite visible quality degradation, as seen from images of Fig 1. We can see that, many edge details are lost, while smooth details are maintained well. This can be seen from the zoomed in images Fig 1(b) and Fig 1(d) where the edges around the cameraman appear 'noisy' in the JPEG image. Further the block-like shape of the distortion could be attributed to the DCT that acts upon non-overlapping blocks of the image, causing discontinuities at block boundaries.



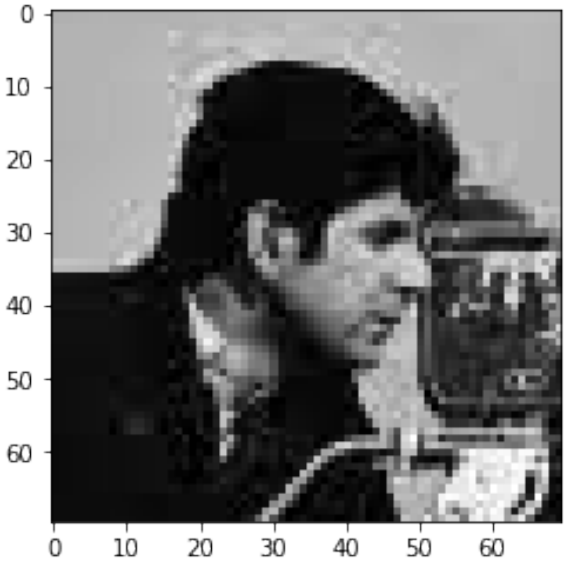
(a) Original Image



(b) Zoomed Original Image



(c) Reconstructed Image



(d) Zoomed Reconstructed Image

Figure 1: JPEG Output Images based on Quantization Matrix

1.2 Results for sub question 2

For the second sub question, where the quantized index of the DCT coefficient $x(i,j)$ is given by :

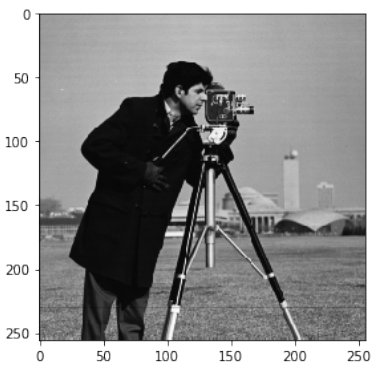
$$y(i,j) = round\,(x(i,j)) \tag{5}$$

we have the following observations:

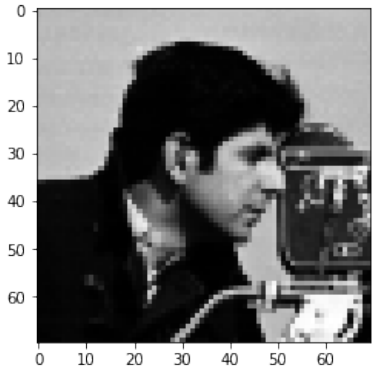
Table 2: Observations for sub-question 1

MSE between original and reconstructed image	**Compression Ratio**	**File Size (in bits)**
0.49	0.99	981341

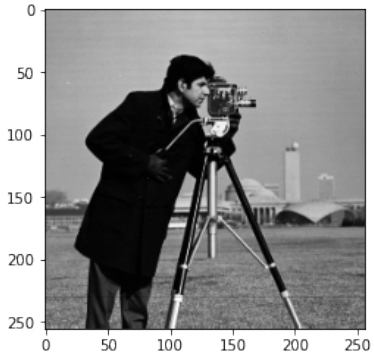
The representation is very close to the original and the reconstructed image has a very low MSE. Which implies that most DCT coefficients are close to true values. Consequently we can also notice, that more non-zero DCT coefficients implies more bits in the bit-stream, hence compression is poor, reducing file size by only **0.99** times the original. Basically it's not even compressing! Visually, the reconstructed image is almost identical to the input images seen in Fig 2(a) and Fig 2(c).



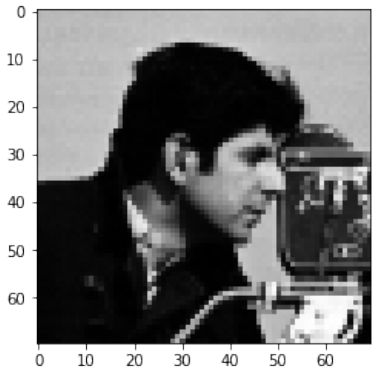
(a) Original Image



(b) Zoomed Original Image



(c) Reconstructed Image



(d) Zoomed Reconstructed Image

Figure 2: JPEG Output Images based on rounding of DCT coefficient

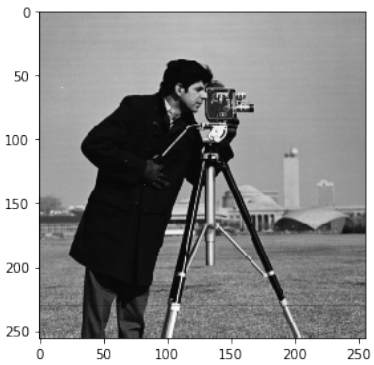
1.3 Results for sub question 3

Here, we implement the JPEG algorithm, and find the optimal values of a,b and c compared to the results of sub-question 1. We are looking for the (a, b, c) that gives the lowest MSE, while giving a compression better than that of sub question 1. The optimal values are $(a^*, b^*, c^*) = (30, 30, 50)$. Also, we have the following observations:

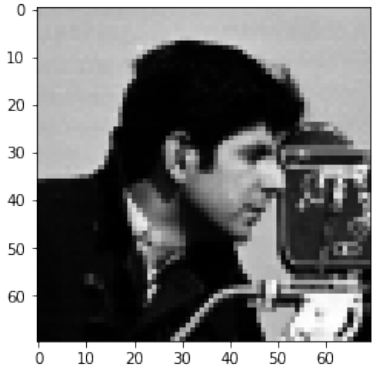
Table 3: Observations for sub-question 1

MSE between original and reconstructed image	**Compression Ratio**	**File Size (in bits)**
26.10	9.77	100392

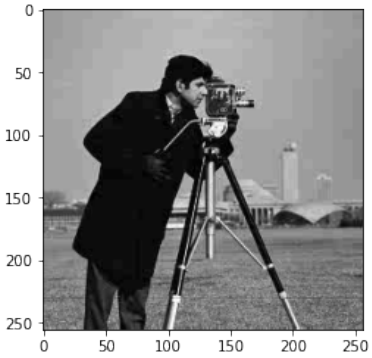
We can see that these values are in accordance which the structure of the DCT matrix. Large value of (a, b, c) corresponds to maximum reduction in number of bits of representation, hence, gives high compression. The MSE will increase during reconstruction for very large values of (a, b, c). Hence, the results we got seem to be a good trade-off between the compression required and maximum MSE admissible. The output reconstructed images are as shown below in Fig 3(c), and looks similar to the reconstructed image seen in Fig 1(c), with the errors near the edges being slightly better than before(sub-question 1).
Also the MSE for sub-question (1) **27.85** is higher than sub-question (3) (optimal a,b,c) **26.10**.



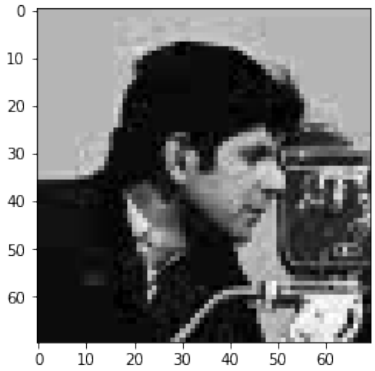
(a) Original Image



(b) Zoomed Original Image



(c) Reconstructed Image



(d) Zoomed Reconstructed Image

Figure 3: JPEG Output Images based on optimal (a, b, c)

2 2-Bit Quantization

We are provided with laplacian source whose pdf is:

$$f_x(x) = \frac{e^{-\frac{|x|}{3}}}{6} \tag{6}$$

2.1 2-bit Scalar Uniform Quantizer

For a 2-bit scalar uniform quantizer with laplacian source, we define distortion as:

$$E[(x-\hat{x})^2] = 2\left(\int_0^{\Delta}\left(x-\frac{\Delta}{2}\right)^2 f_X(x)dx + \int_{\Delta}^{\infty}\left(x-\frac{3\Delta}{2}\right)^2 f_X(x)dx\right) \tag{7}$$

Here I have used the symmetric property of integral and compressed the four integrals into two. Now, to get the optimum Δ, we must differentiate (7) w.r.t Δ and put it equal to zero. We get the following,

$$\frac{d}{d\Delta}E[(x-\hat{x})^2] = \frac{1}{4}[(\Delta^2 - (\Delta^2+12\Delta+72)e^{\frac{-\Delta}{3}} - 12\Delta + 72) + ((\Delta-12)\Delta+72)e^{\frac{-\Delta}{3}}] = 0 \tag{8}$$

Solving (8), we get:

$$\Delta^* = 3W\left(\frac{8}{e^2}\right) + 6 \tag{9}$$

Where, $W(.)$ is the Lambert W-function, also called the omega function, is the inverse function of:

$$f(W) = We^W \tag{10}$$

From (8) and (9) we can say that the optimal Δ can be found in an iterative manner. Hence, to find optimal Δ we performed a line search for **100** values of delta starting from **0.01** to **10**. For each delta, we computed the distortion by evaluating the integral given in (7) using scipy's function.
We found our optimal Δ^* in this case to be **4.65** and the distortion corresponding to it is **3.53**. Below figure shows the total distortions w.r.t Δ for 2-bit uniform quantizer of laplacian source obtained by numerical simulations (explained above).

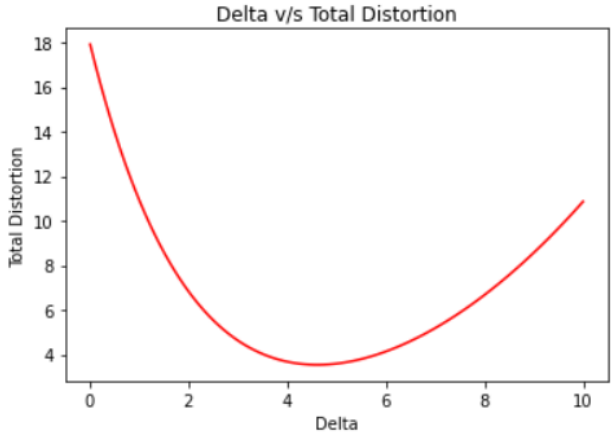


Figure 4: Delta v/s Total Distortion Plot

The decision boundaries are : $[-\infty, -4.65, 0, 4.65, \infty]$ and the optimal quantization labels for this problem is: $[-6.97, -2.32, 2.32, -6.97]$.

2.2 2-bit Lloyd-Max Quantizer

The following iterative algorithm is implemented to find optimal quantization labels and decision boundaries:

- Guess initial set of representative levels $\hat{x}_q$. Where $q \in \{0,1,...M\text{-}1\}$. Here $M = 4$.
- Compute the decision boundaries as:
$$t_q = 0.5(\hat{x}_{q-1} + \hat{x}_q) \tag{11}$$
- Calculate new quantization labels:
$$\hat{x}_q = \frac{\int_{t_q}^{t_{q+1}} x.f_X(x)dx}{\int_{t_q}^{t_{q+1}} f_X(x)dx} \tag{12}$$
- Repeat the second and third step until no further distortion reduction beyond a very small value ϵ. I have taken ϵ to be $\mathbf{10}^{-15}$.

The initial set of quantization labels are $[-3, -1, 1, 3]$. We compute the distortion $\mathbf{D}$ achieved by Lloyd-Max quantizer using:

$$D = \int_{-\infty}^{t_1} (x - \hat{x}_0)^2 f_X(x)dx + \int_{t_1}^{t_2} (x - \hat{x}_1)^2 f_X(x)dx + \int_{t_2}^{t_3} (x - \hat{x}_2)^2 f_X(x)dx + \int_{t_3}^{+\infty} (x - \hat{x}_3)^2 f_X(x)dx \tag{13}$$

I have done **100** iterations of the above algorithm. All the integrations involved in this part are done using numerical integration methods using scipy. The decision boundaries are : $[-\infty, -4.78, 0, 4.78, \infty]$ and the optimal quantization labels for this problem is: $[-7.78, -1.78, 1.78, 7.78]$. The distortion with the optimal parameters for Llyod-Max quantizer is **3.1**.

2.3 Comparison of Llyod-Max and Uniform Quantizer

The reason that Llyod-Max is giving less distortion as it's trying to achieve minimum distortions by iterating over decision boundaries and quantization labels. So, basically we're adjusting the decision boundaries and quantizations labels such that distortion will be minimum. In uniform quantizer we allot some sort of a specific structure to the decision boundaries and quantization labels, hence it will give more distortion than Llyod-Max.

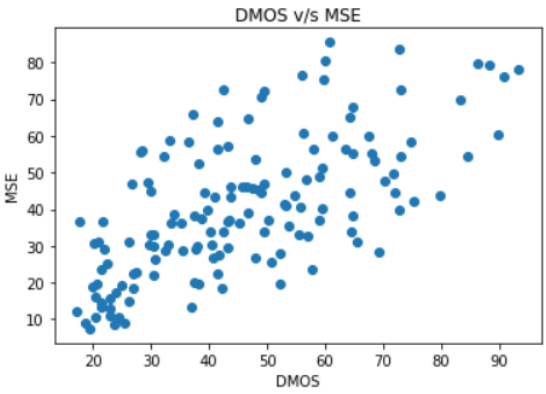
Table 4: Comparison of Uniform & Llyod-Max Quantizer

Distortion	**Decision Boundary**	**Quantization Labels**
3.53 (Uniform)	[-$\infty, -4.65, 0, 4.65, \infty$]	[-6.97, -2.32, 2.32, 6.97]
3.10 (Llyod-Max)	[-$\infty, -4.78, 0, 4.78, \infty$]	[-7.78, -1.78, 1.78, 7.78]

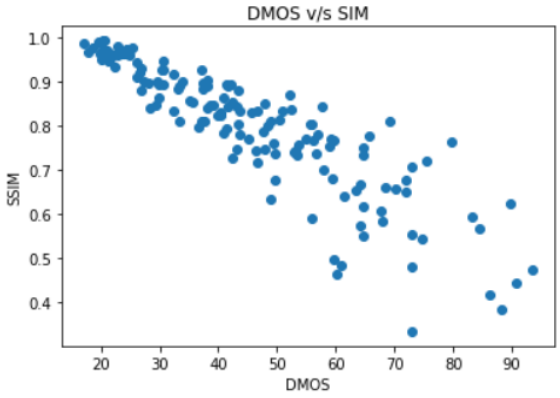
3 Image Quality Assessment

Upon implementing the algorithm we see that the MSE and SSIM values vary vastly over the blurred images based on blurring. The lowest MSE of **7.46** corresponds to an image with SSIM **0.98**, while that with the Highest MSE of **85.56** has an SSIM of **0.48**, which is close to the lowest MSE observed. These are good results, considering the visual quality of the images, and we can see that the SSIM value is able to accurately represent these variations as well. On the contrary, an image with highest SSIM of **0.99** has an MSE of **10.77**, which is good, but for the image with lowest SSIM **0.33**, the MSE is **83.54**. From looking at the images, we can also say that there is no clear threshold for the MSE or SSIM which can differentiate visually appealing and unappealing Images, but we can measure how well they approximate to human rating.

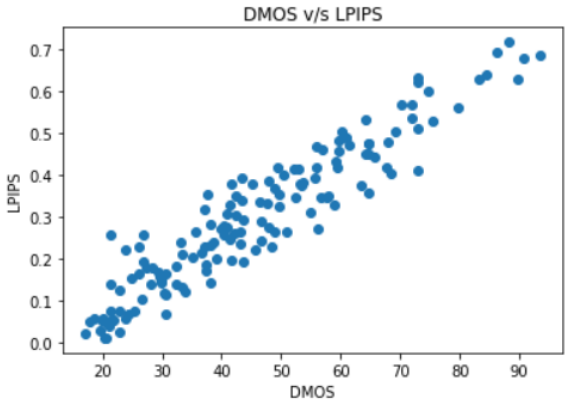
3.1 Results



(a) Scatter plot of DMOS v/s MSE scores



(b) Scatter plot of DMOS v/s SSIM scores



(c) Scatter plot of DMOS v/s LPIPS scores

Figure 5: Scatter plots for Problem 3

3.2 Performance Comparison

We can clearly see from the plots that MSE doesn't shows strong correlation with DMOS scores, while SSIM and LPIPS does. LPIPS shows the strongest correlation with DMOS. SSIM starts de-correlating for high DMOS values and low SSIM values, i.e **for very blurred image SSIM may pop out to be moderate**.
Why LPIPS performs best ? In neurobiology, there is a concept called "lateral inhibition". Now what does that mean? This refers to the capacity of an excited neuron to subdue its neighbors. We basically want a significant peak so that we have a form of local maxima. This tends to create a contrast in that area, hence increasing the sensory perception. Increasing the sensory perception is a good thing! We want to have the same thing in our CNNs. LPIPS gives excellent result because it extracts deep features from pre-trained VGG network, in VGG there's something called Local Response Normalization that inhibits the lateral inhibition. Hence mimicing human perceptual system !

The following are the Spearman's Correlation Coefficient for various metrics used:

$$\rho_{MSE} = 0.65$$

$$\rho_{SSIM} = -0.91$$

$$\rho_{LPIPS} = 0.93$$

The negative sign just indicates that the relation is inverse, with SSIM decreasing as DMOS increases.