Problem 1: JPEG Implementation

Objective:

The objective of this problem is to implement a toy version of JPEG on the given image and observe the level of compression for various Quantization matrices

Implementation Details:

- The given grayscale image is divided into 8 x 8 blocks and Discrete Cosine Transform is applied on each of the blocks to convert the pixel values from spatial domain to frequency domain
- From the given 8 x 8 Quantization matrix, the DCT coefficients are quantized using the given equation so that the amount of data needed to represent them reduces.

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

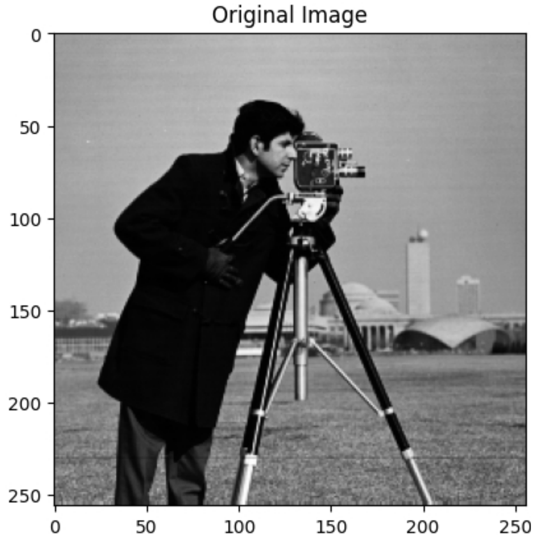
- The quantized DCT coefficients (blockwise) are then encoded using Huffman coding to furthur reduce the data representation. The Huffman coding scheme assigns variable length codes to each quantized coefficient based on their frequency of occurance
- The quantized DCT coefficients are dequantized (blockwise) using the Quantization matrix in the following manner

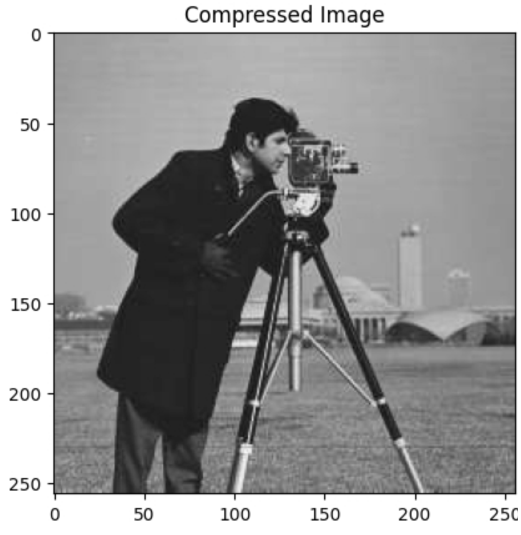
$$\hat{x}(i,j) = y(i,j)Q(i,j)$$

- Inverese DCT is applied (blockwise) to transform them back to spatial domain and the reconstructed 8 x 8 blocks are assembled back into full image
- To observe the level of compression and file size, Reconstruction of Image is done with many scaled versions of Quantization matrix (It is scaled up with all integers between 50 – 1 and scaled down with the same integers)

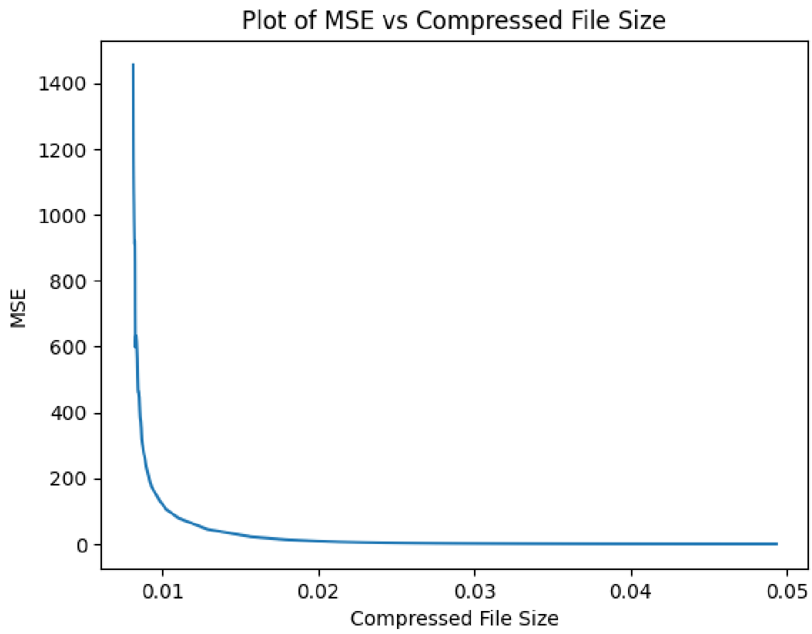
Results and Observations:

The following are the results obtained during implementation:





```
Input Image Size in MB =  0.0625
Compressed Image Size in MB =  0.012904882431030273
Compression Ratio = 4.84312819849613
MSE between Original and Compressed Images = 44.54563508062465
```



- When the quantization matrix is scaled up (i.e., the quantization factor is increased), the quantized DCT coefficients will have larger values, which means that more coefficients will be rounded to zero during quantization. This leads to a higher level of compression, as fewer bits are needed to represent the quantized coefficients. However, increasing the quantization factor also leads to higher lossy distortion, as the higher values of the quantized coefficients are more likely to contain important image details. Thus **MSE increases and filesize decreases when Quantization matrix is scaled up**
- On the other hand, when the quantization matrix is scaled down (i.e., the quantization factor is decreased), the quantized DCT coefficients will have smaller values, which means that more coefficients will be retained and fewer coefficients will be rounded to zero during quantization. This leads to a lower level of compression, as more bits are needed to represent the quantized coefficients. However, decreasing the quantization factor also leads to lower lossy distortion, as the lower values of the quantized coefficients are less likely to contain important image details. Thus **MSE decreases and file size increases when Quantization matrix is scaled down**

Problem 2: Bit allocation for uniform scalar quantization of independent sources

Consider a pair of uniformly distributed continuous independent sources X_1 and X_2, both with mean zero and variances 5 and 10 respectively. Let the rate of the uniform scalar quantizers used for X_1 and X_2 be R_1 and R_2 respectively where $R_i = \log_2 K_i$ for $i \in \{1, 2\}$ and K_i denotes the number of quantization points. Given a budget sum rate constraint of $R_1 + R_2 = 3$, compute the rate pair (R_1, R_2) that minimizes the sum of squared error in the respective reconstructions $\mathbb{E}\left[(X_1 - \hat{X}_1)^2\right] + \mathbb{E}\left[(X_2 - \hat{X}_2)^2\right]$. Comment on how you would allocate bits among different DCT coefficients based on this result.

The rate pair (R1, R2) that minimizes the sum of squared error in the respective reconstructions is (1, 2). Since this is uniform scalar quantization, Rates and Quantization levels have to be integers.

We have to allocate bits to different DCT coefficients in such a way that the total distortion in the reconstructed image is minimized subject to the constraint that the bit rates sum to 3.

Thus , we can allocate **more bits** to the DCT coefficients that correspond to the **independent source with higher variance**, since these coefficients are likely to have more significant information content and **less bits** to the DCT coefficients that correspond to i**ndependent source with lower variance**, since these coefficients are less likely to have more significant information content

Solution is given in the next page

Problem - 2 :

$X_1, X_2 \rightarrow$ Uniformly distributed continuous independent sources ~~X_1~~ and

$\mu_{X_1} = \mu_{X_2} = 0 \qquad \sigma^2_{X_1} = 5 \qquad \sigma^2_{X_2} = 10.$

Rates of Uniform scalar quantizers used for X_1, X_2 are R_1 & R_2 respectively.

$R_i = \log_2 K_i$ for $i \in \{1,2\}$ where K_i denotes the no. of quantization points

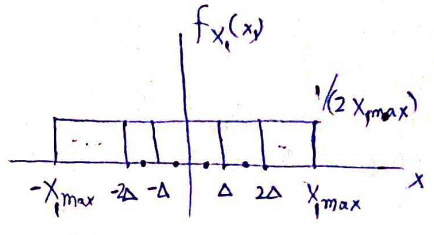
Given $R_1 + R_2 = 3$.

$\therefore$ The following feasible (R_1, R_2) are $(1,2)$ and $(2,1)$, since no. of Quantization point and rates are integers in case of Uniform scalar quantization.

$\Rightarrow$ Now we have to choose (R_1, R_2) such that $\mathbb{E}[(X_1-\hat{X}_1)^2] + \mathbb{E}[(X_2-\hat{X}_2)^2]$ is minimum

Let $\hat{X}_1 = Q(X_1)$.

$$\mathbb{E}[(X_1-\hat{X}_1)^2] = \int_{-X_{1max}}^{X_{1max}} [x_1 - Q(x_1)]^2 f_{X_1}(x_1)\, dx_1$$



$$= 2 \underbrace{\sum_{i=1}^{K_1/2} \int_{(i-1)\Delta_1}^{i\Delta_1} (x_1 - (i - 1/2)\Delta_1)^2 \left[\frac{1}{2X_{1max}}\right] dx_1}_{\text{each of the integrals is identical}}$$

$$\Delta_1 = \frac{2X_{1max}}{K_1}$$

$$= 2 \cdot \frac{K_1}{2} \int_{-\Delta/2}^{\Delta/2} q^2 \left[\frac{1}{2X_{max}}\right] dq$$

$$= \int_{-\Delta/2}^{\Delta/2} q^2 \cdot \frac{1}{\Delta_1} \cdot dq$$

$$= \frac{\Delta_1^2}{12} = \sigma^2_{X_1} \cdot 2^{-2R_1}$$

$$\therefore \mathbb{E}[(X_1-\hat{X}_1)^2] + \mathbb{E}[(X_2-\hat{X}_2)^2] = \frac{\Delta_1^2 + \Delta_2^2}{12} = \sigma^2_{X_1} \cdot 2^{-2R_1} + \sigma^2_{X_2} \cdot 2^{-2R_2}$$

Distortion at $(R_1, R_2) = (1,2)$ is $= 5 \cdot 2^{-2} + 10 \cdot 2^{-4} = 1.875$

Distortion at $(R_1, R_2) = (2,1)$ is $= 5 \cdot 2^{-2\times 2} + 10 \cdot 2^{-2} = 2.8125$

The Rate pair (R_1, R_2) that minimizes the Distortion is $(1,2)$

Problem 3: YOLO Object Detection

Objective:

The Objective of this problem is to train a YOLO Object Detection model on the given dataset and evaluate the performance of the trained model on the given test set

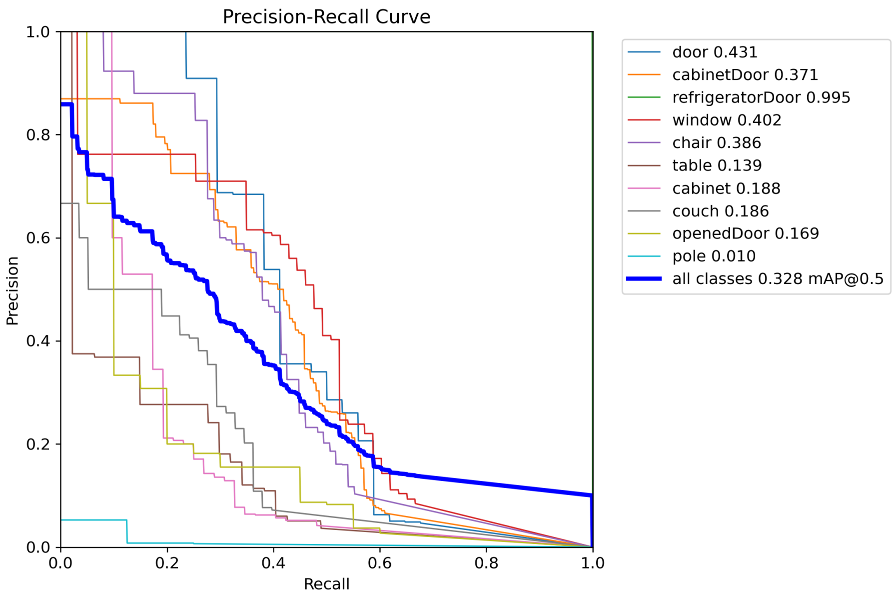
Implementation Details:

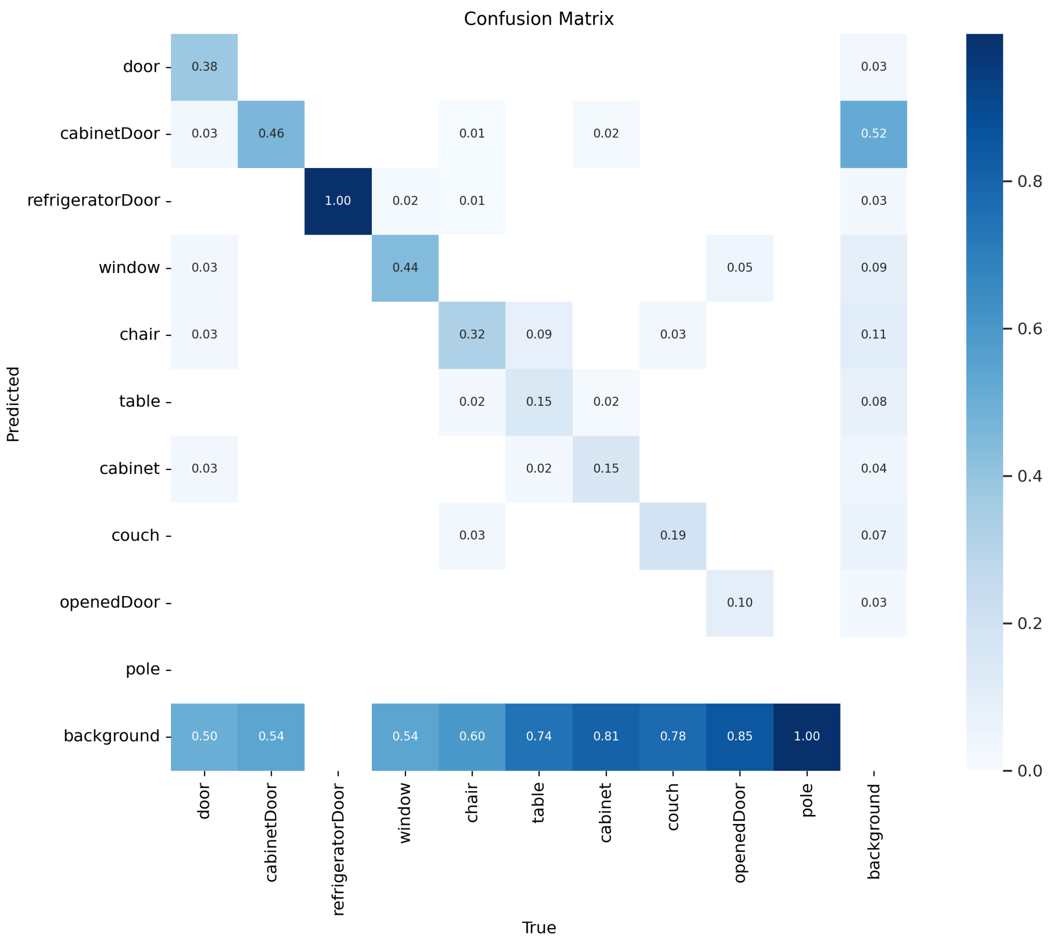
- YOLOv5s model is chosen for Object Detection and the analysis is done as per this document https://docs.ultralytics.com/yolov5/train_custom_data/
- After importing the model and installing the requirements, the given data.yaml is edited so that the directories of train, test and validation dataset are modified as per required
- Initially, the model is initialized with pretrained weights and then trained on the given train data for 100 epochs and then tested on the given test data
- Later on, the model is initialized with random weights and then trained on the given train data for 100 epochs and then tested on the given test data

Results and Observations:

Analysis on test data when model is trained with ***pretrained weight initializations****:*

Class	Images	Instances	Precision (P)	Recall (R)	mAP50	mAP50-95
all	**107**	**550**	**0.52**	**0.317**	**0.328**	**0.173**
door	107	34	0.641	0.382	0.431	0.185
cabinetDoor	107	179	0.486	0.419	0.371	0.143
refrigeratorDoor	107	2	0.237	1	0.995	0.646
window	107	63	0.601	0.413	0.402	0.237
chair	107	87	0.578	0.33	0.386	0.192
table	107	47	0.35	0.149	0.139	0.0608
cabinet	107	52	0.506	0.173	0.188	0.115
couch	107	58	0.44	0.203	0.186	0.08
openedDoor	107	20	0.366	0.1	0.169	0.0696
pole	107	8	1	0	0.0101	0.00341

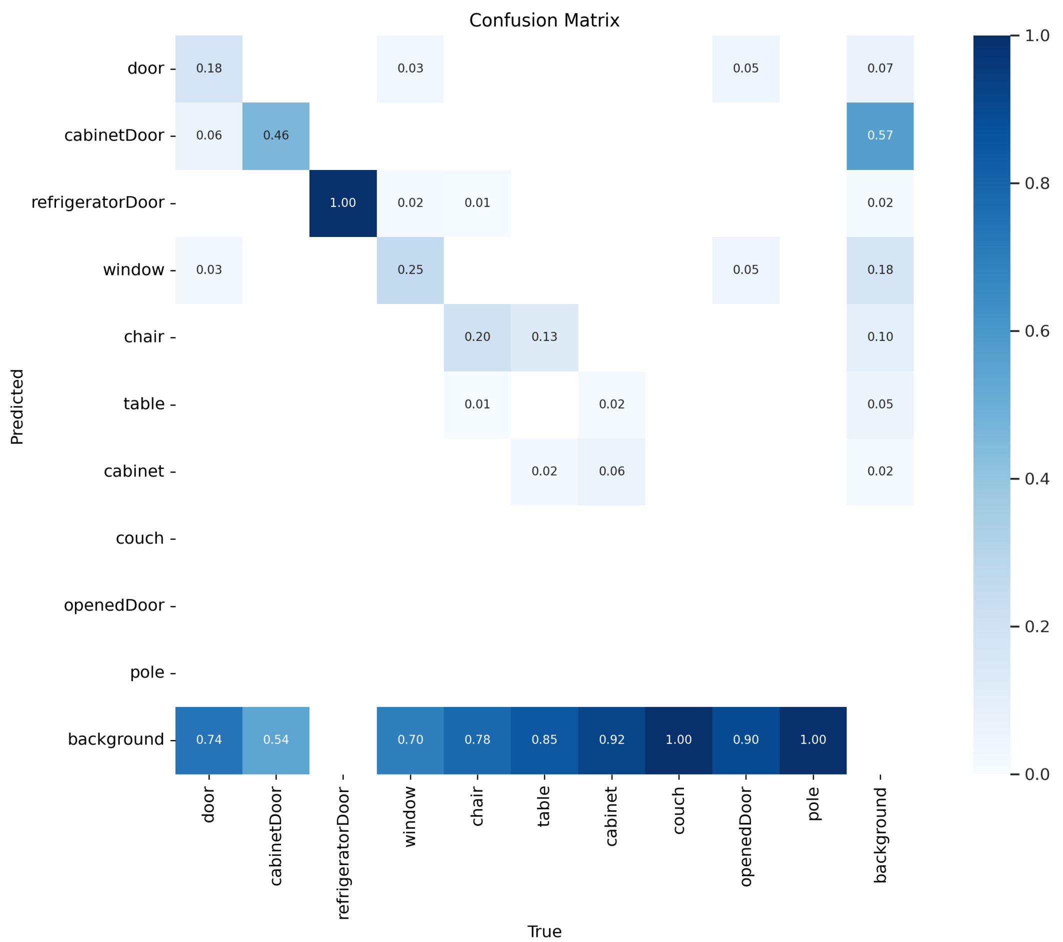


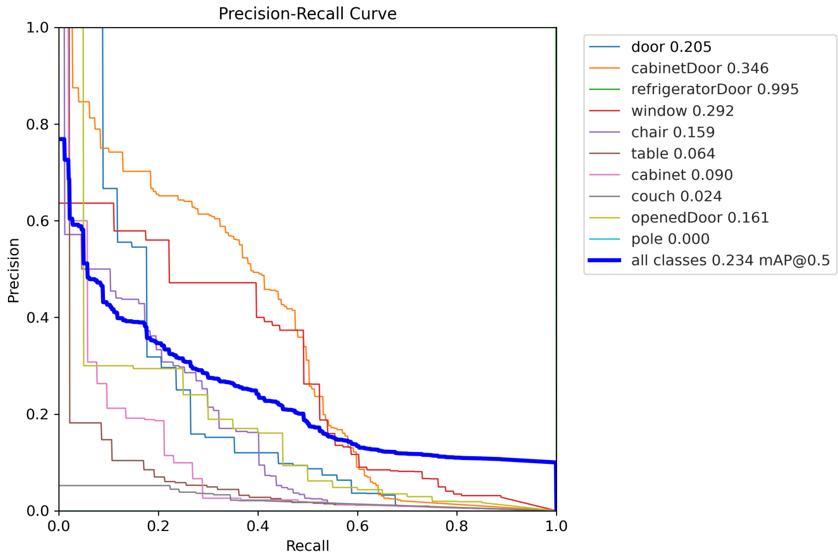

Confusion Matrix
Predicted
True
door
cabinetDoor
refrigeratorDoor
window
chair
table
cabinet
couch
openedDoor
pole
background
0.0
0.2
0.4
0.6
0.8


1003.png
126.png
1290.png
124.png
1253.png
1239.png
1281.png
1232.png

*Analysis on test data when model is trained with **random weight initializations**:*

Class	Images	Instances	Precision (P)	Recall (R)	mAP50	mAP50-95
all	**107**	**550**	**0.499**	**0.243**	**0.234**	**0.111**
door	107	34	0.256	0.235	0.205	0.0896
cabinetDoor	107	179	0.401	0.475	0.346	0.118
refrigeratorDoor	107	2	0.183	1	0.995	0.596
window	107	63	0.421	0.397	0.292	0.136
chair	107	87	0.297	0.207	0.159	0.0553
table	107	47	0.125	0.0426	0.0641	0.0215
cabinet	107	52	0.307	0.0768	0.0896	0.038
couch	107	58	1	0	0.0237	0.00645
openedDoor	107	20	1	0	0.161	0.0486
pole	107	8	1	0	0	0





- We can note that the mAP50 and mAP50-95 values are better with pretrained weight initializations than that of random initialization. By using the pretrained weights as initialization, the object detection model can start with weights that have already learned useful features for detecting objects, such as edges, corners, and textures. This allows the model to more quickly learn the specific features needed for detecting objects in a new dataset. This initialization helps the model converge faster and achieve better performance.
- The model when initialized in both ways shown no significant performance improvement during training after 100 epochs. This could be because the train data is insufficient and the model couldnt learn enough features to detect objects