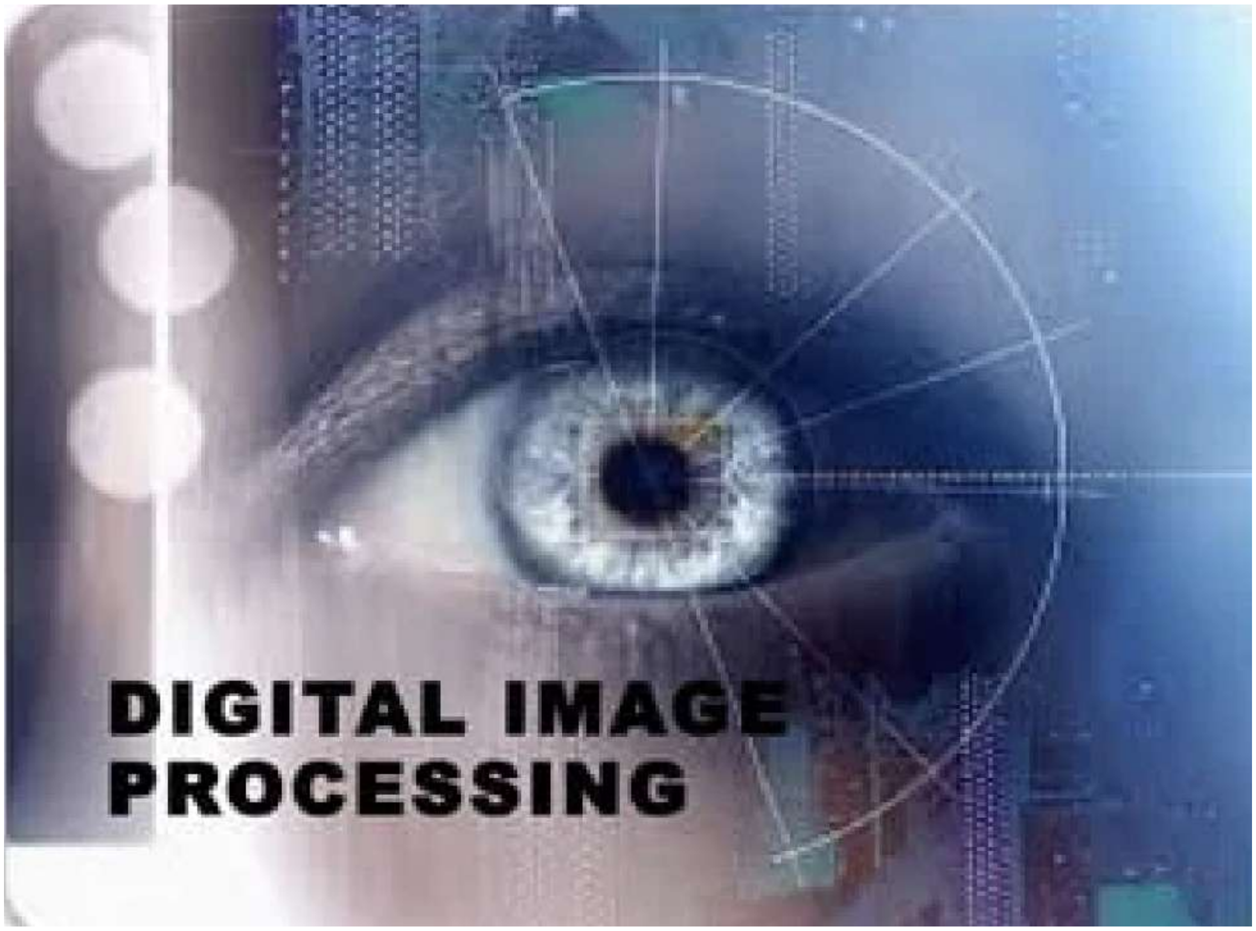
Digital Image Processing

Harshit Gupta(2021CSB1092)

JPEG Compression



Introduction

JPEG (Joint Photographic Experts Group) is a popular image compression format that uses lossy compression techniques to reduce the file size of images while preserving visual quality. It was created in the late 1980s by the Joint Photographic Experts Group, which was a committee of experts from industry and academia tasked with developing a standard for compressing digital images.

JPEG compression works by dividing an image into small blocks of pixels, which are then transformed using a mathematical algorithm called the Discrete Cosine Transform (DCT). The resulting coefficients are then quantized, which means that some of the coefficients are rounded to zero, reducing the amount of data in the image. The quantized coefficients are then encoded using variable-length coding, which assigns shorter codes to more frequently occurring coefficients. The resulting compressed data can be stored in a JPEG file, which can be decompressed to recreate the original image.

JPEG compression is widely used for compressing digital images, especially photographs, for web publishing, email attachments, and other digital media applications. It can achieve high compression ratios with relatively little loss of visual quality, making it an ideal choice for situations where file size is a concern. However, it is a lossy compression technique, which means that some data is lost during the compression process. The amount of data loss can be controlled by adjusting the compression level.

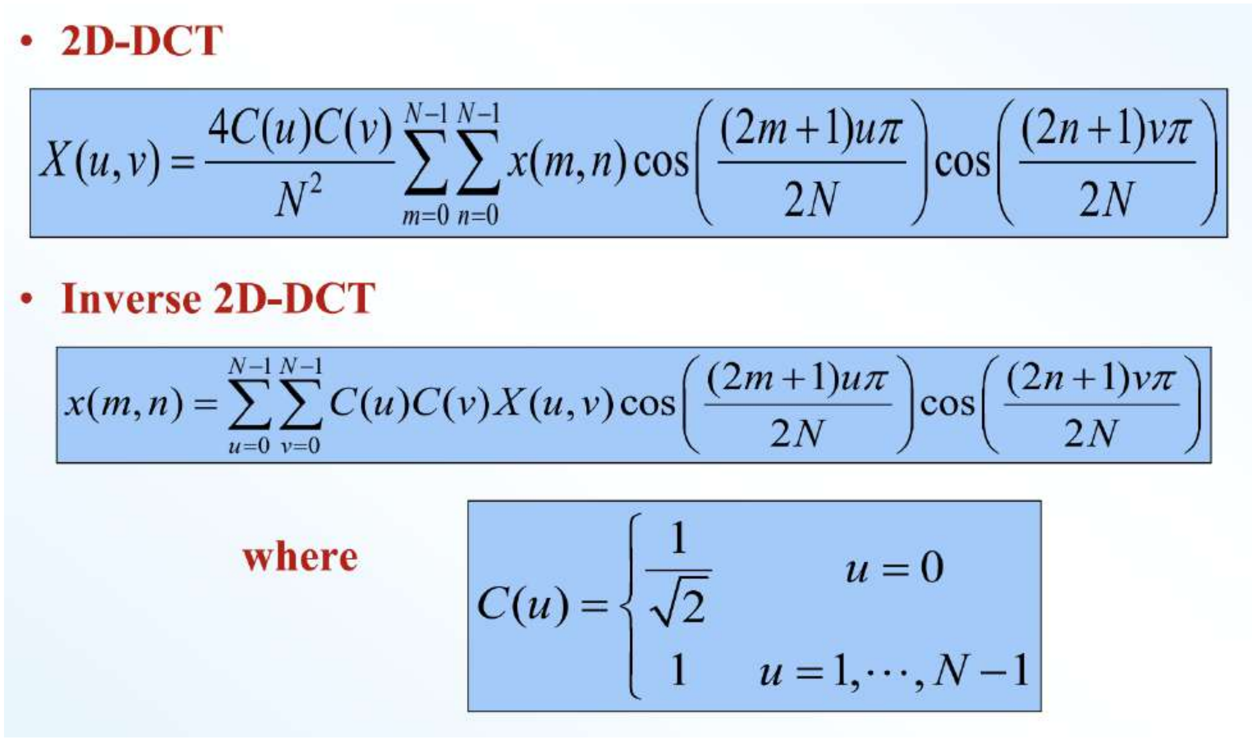
Some Common Terms

Compression Ratio

It represents the amount by which the original data is compressed, expressed as a ratio of the size of the original data to the size of the compressed data.
In the case of JPEG compression, the compression ratio can range from 2:1 to 100:1 depending on the compression level. For example, a compression ratio of 10:1 means that the compressed data is one-tenth the size of the original data. The higher the compression ratio, the more the data is compressed, but also the more lossy the compression process becomes.

DCT(Discrete Fourier Transform)

DCT stands for Discrete Cosine Transform, which is a mathematical algorithm used in JPEG compression to transform an image from the spatial domain to the frequency domain. This means that the image is represented in terms of its frequency components rather than its pixel values.
The DCT works by breaking down the image into small blocks of pixels, typically 8x8 pixels, and then applying a mathematical formula to each block.
The DCT is a reversible process, which means that the original image can be reconstructed from the frequency domain representation by applying the inverse DCT. However, during JPEG compression, some of the coefficients in the frequency domain representation are discarded, leading to some loss of information in the image.



- **2D-DCT**

$$X(u,v)=\frac{4C(u)C(v)}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x(m,n)\cos\left(\frac{(2m+1)u\pi}{2N}\right)\cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

- **Inverse 2D-DCT**

$$x(m,n)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)X(u,v)\cos\left(\frac{(2m+1)u\pi}{2N}\right)\cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

where

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}} & u=0\\ 1 & u=1,\cdots,N-1\end{cases}$$

Quantization

Quantization is a process used in JPEG compression to reduce the amount of data in an image by rounding off some of the coefficients in the frequency domain. The amount of quantization applied depends on the compression level selected. Quantization works by dividing each coefficient by a fixed quantization factor and then rounding the result to the nearest integer value
The quantization process in JPEG compression is lossy, meaning that some information is lost during the process.

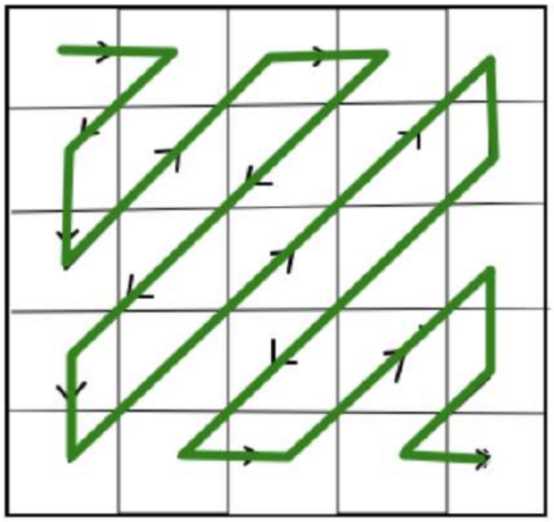
Here we have used an standard quantization matrix of size[8*8] for quantization and we are resizing the quantization matrix when the block size given is different. The quantization matrix that is used in the code is given below

```
[[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]]
```

Zigzaging

After the DCT process and quantization, the resulting coefficients are arranged in a matrix that represents the frequency domain representation of the 8x8 block. However, the coefficients are not ordered in a way that is optimal for entropy coding, which can result in larger compressed file sizes.

Zigzagging involves reordering the coefficients in a zigzag pattern that starts at the top-left corner and ends at the bottom-right corner of the matrix. This reordering results in a sequence of coefficients that have a high probability of having many consecutive zeros,that we have discarded here.



PSNR

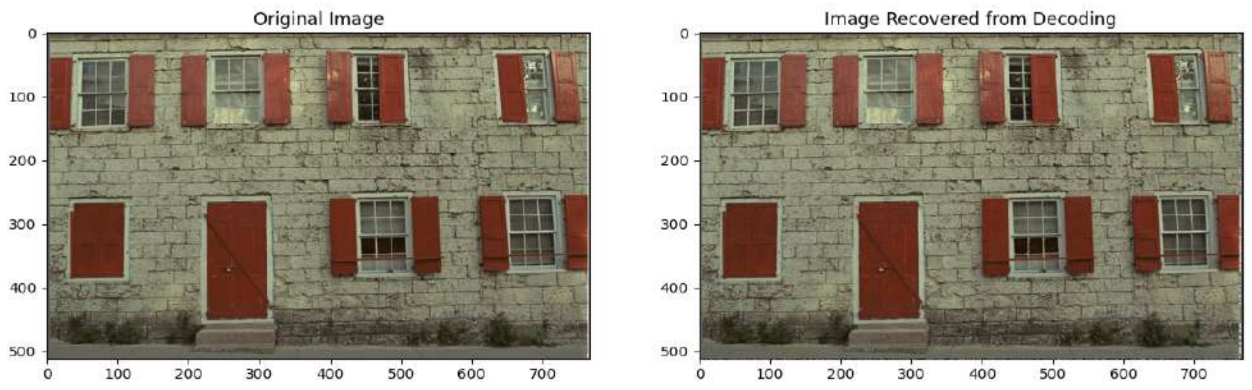
It is a quantitative measure that provides a numerical value to represent the difference between the original uncompressed image and the compressed image. PSNR is calculated by comparing the pixel values of the original and compressed images and measuring the difference in terms of the root mean squared error (RMSE)

PSNR = 20 * log10((MAX) /RMSE)

where MAX is the maximum possible pixel value (e.g., 255 for an 8-bit image).

Results

The Jpeg compression gives different types of results when you give different parameters. When we run the code it asks for 4 different parameters. The input image,whether the image is coloured or grayscale,Block Size,Number of coefficients(the number of terms for array of each block).The image below shows the result when our block size is 8*8 and we are passing complete compressed array after removing zeros from the end and the image is coloured.

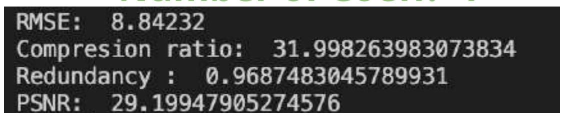


Variation of number of coefficients:-

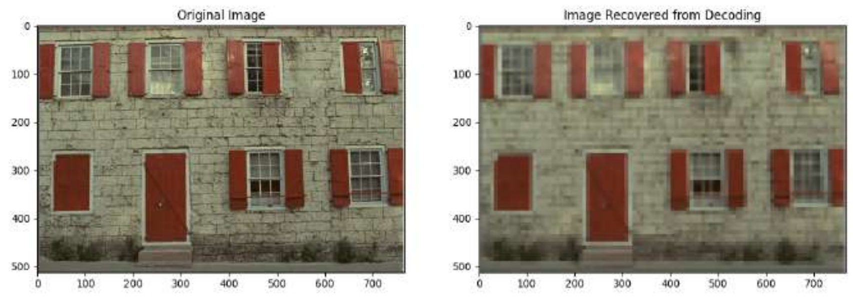
As we increase the number of coefficients, our PSNR increases as RMSE is decreasing ,the image will look more like the original image ,The compression ratio will decrease as we increase the number of coefficients as the array will become larger in size.

The various results for different number of coefficients for a given image is shown below.The block size is taken as 8*8 here.

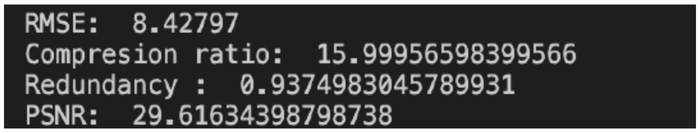
Number of Coeff.=1



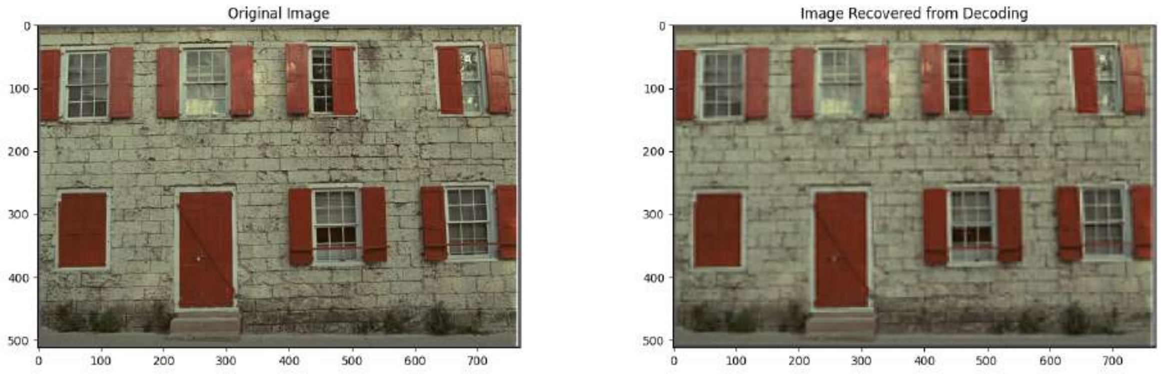
```
RMSE:  8.84232
Compresion ratio:  31.998263983073834
Redundancy :  0.9687483045789931
PSNR:  29.19947905274576
```



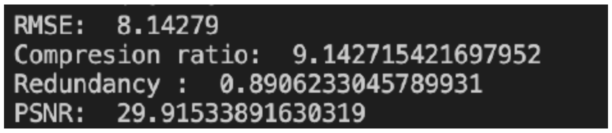
Number of Coeff.=3



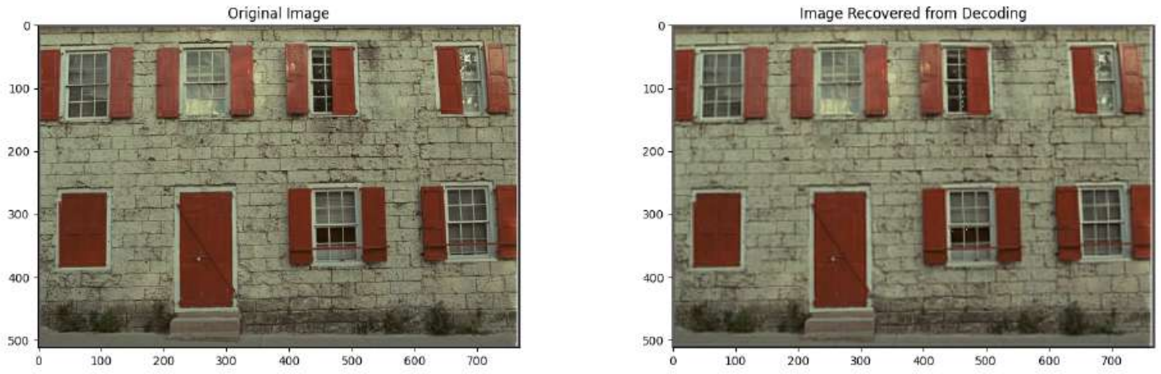
```
RMSE:  8.42797
Compresion ratio:  15.99956598399566
Redundancy :  0.9374983045789931
PSNR:  29.61634398798738
```



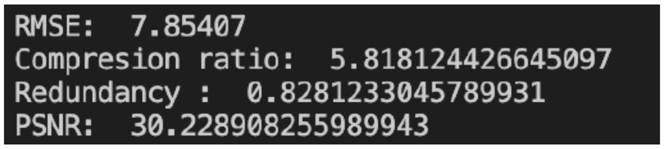
Number of Coeff.=6



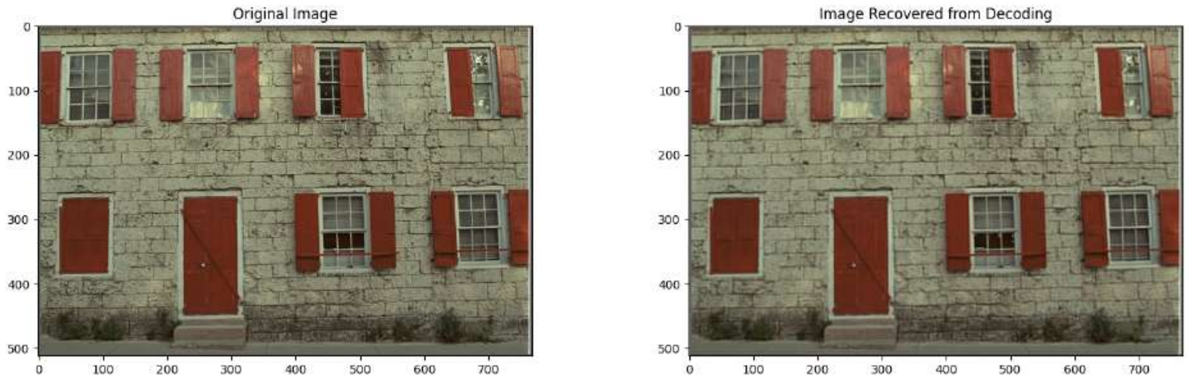
```
RMSE:  8.14279
Compresion ratio:  9.142715421697952
Redundancy :  0.8906233045789931
PSNR:  29.91533891630319
```



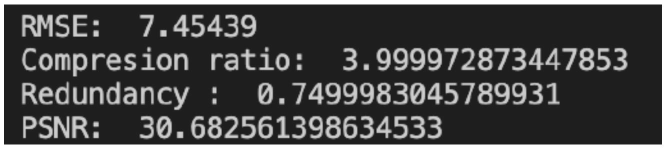
Number of Coeff.=10



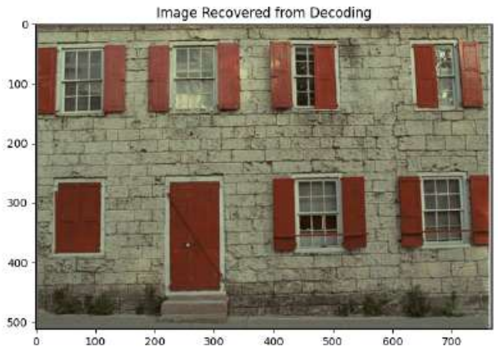
```
RMSE:  7.85407
Compresion ratio:  5.818124426645097
Redundancy :  0.8281233045789931
PSNR:  30.228908255989943
```



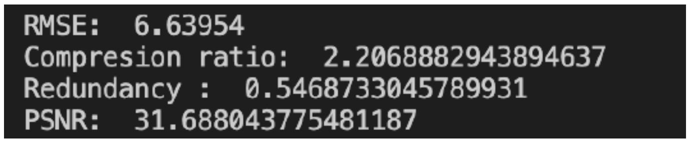
Number of Coeff.=15



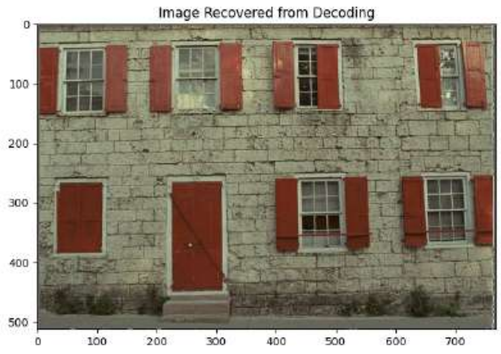
```
RMSE:   7.45439
Compresion ratio:   3.999972873447853
Redundancy :  0.7499983045789931
PSNR:   30.682561398634533
```



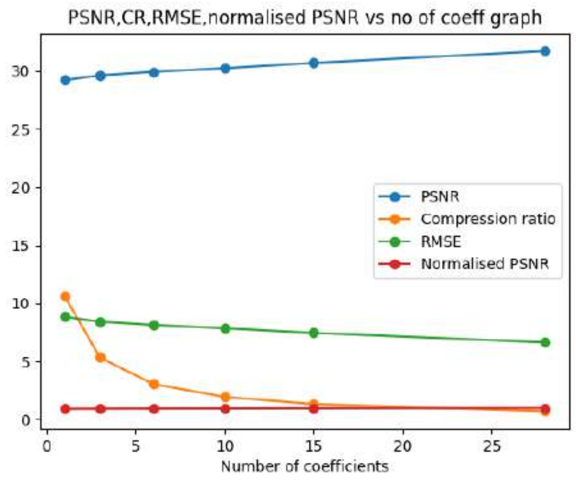
Number of Coeff.=28



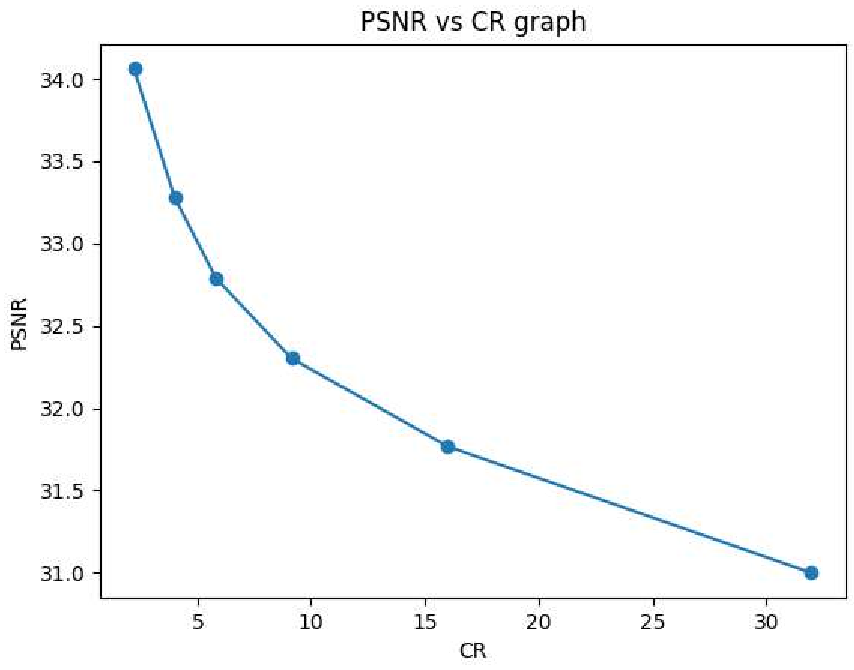
```
RMSE:   6.63954
Compresion ratio:   2.2068882943894637
Redundancy :  0.5468733045789931
PSNR:   31.688043775481187
```



The following graph show the variation of different parameter on this image as the number of coefficients changes when the block size is fixed to 8*8



The following graph show the average PSNR vs average Compression Ratio graph of 24 different images from dataset as the number of coefficients varies:-

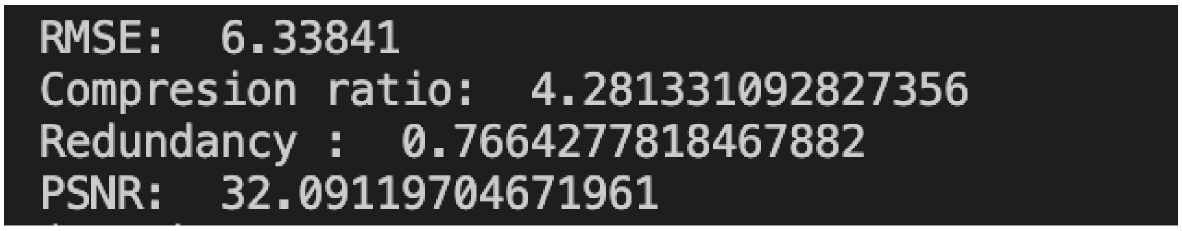


Variation of Block size:-

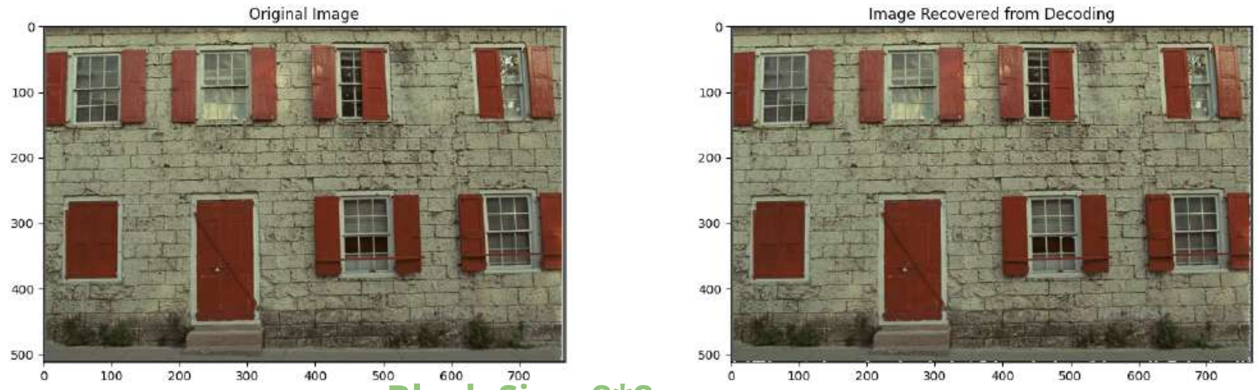
As we increase the block size, our PSNR decreases as RMSE is increasing ,the image will look less like the original image ,The compression ratio will firstly increase and achieve a max value but after that it decreases as we increase the block size.

The various results for different block size for a given image is shown below.Here,we are passing complete compressed array after removing zeros from the end and the image is coloured.

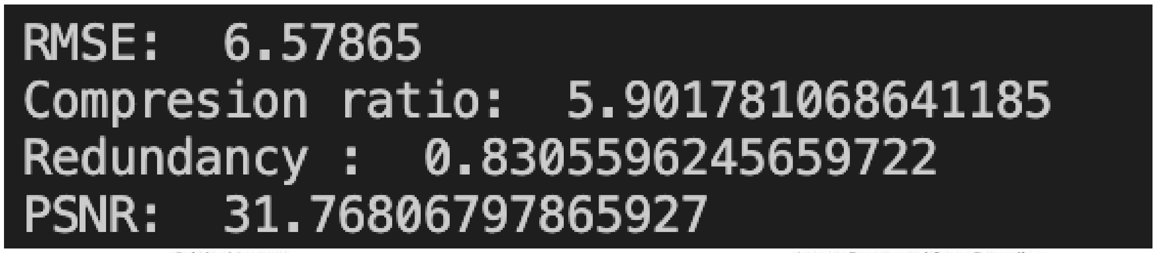
Block Size=4*4



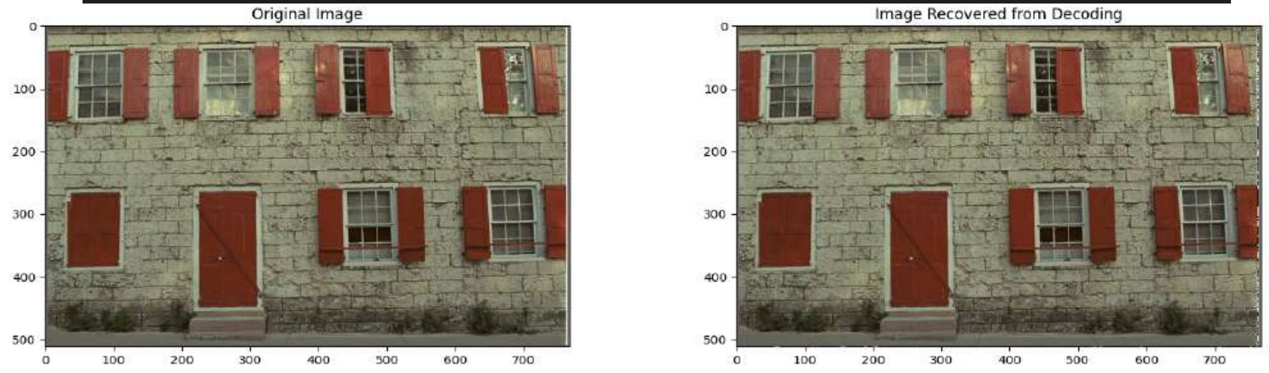
```
RMSE:  6.33841
Compresion ratio:  4.281331092827356
Redundancy :  0.7664277818467882
PSNR:  32.09119704671961
```



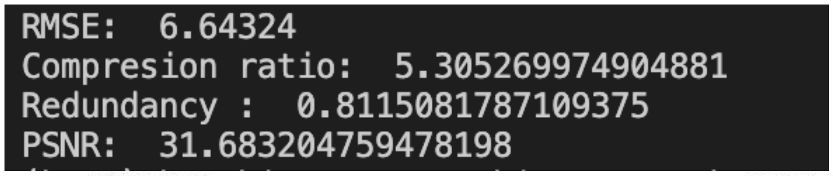
Block Size=8*8



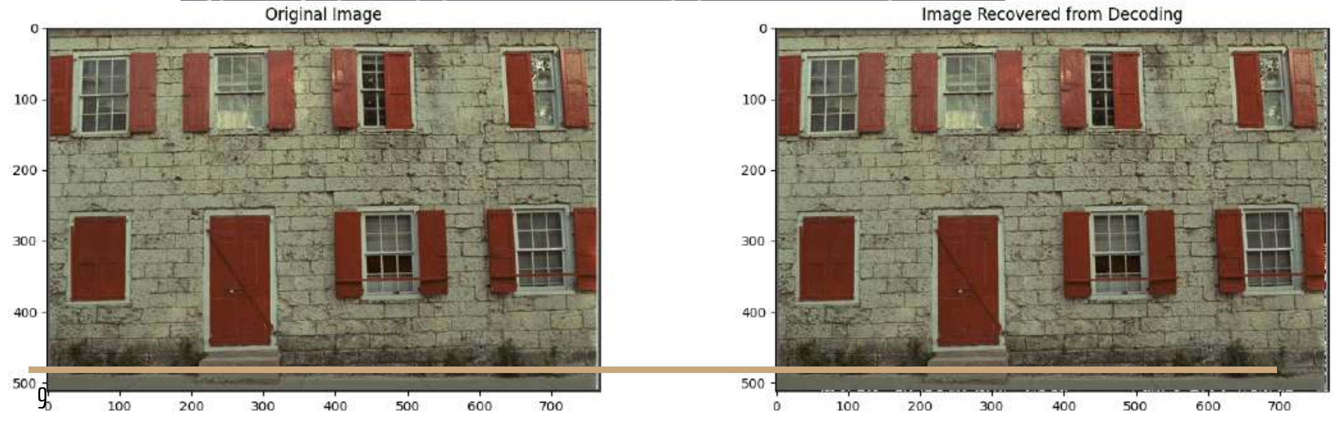
```
RMSE:  6.57865
Compresion ratio:  5.901781068641185
Redundancy :  0.8305596245659722
PSNR:  31.76806797865927
```



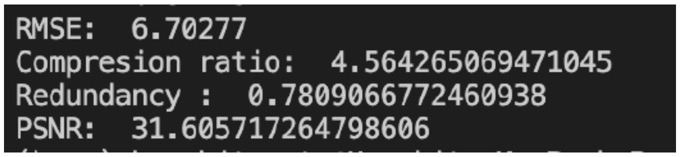
Block Size=16*16



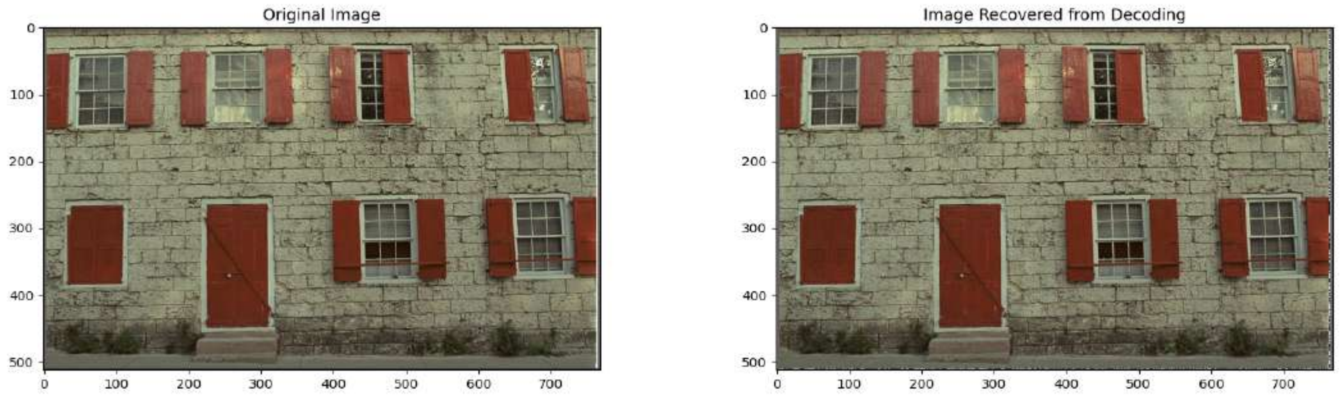
```
RMSE:  6.64324
Compresion ratio:  5.305269974904881
Redundancy :  0.8115081787109375
PSNR:  31.683204759478198
```



Block Size=32*32



```
RMSE:  6.70277
Compresion ratio:  4.564265069471045
Redundancy :  0.7809066772460938
PSNR:  31.605717264798606
```



Observation

We observed the effects of varying number of coefficients on different parameters like PSNR, RMSE, Compression ratio and Normalized PSNR. From the Graph shown in the report above we can say that the PSNR and Normalized PSNR increases as the number of coefficients increases and the Compression ratio and RMSE decreases as the number of coefficients increases.In this case, we have fixed the block size

We observed the effects of varying block size on different parameters like PSNR, RMSE, Compression ratio. From the data shown in the report above we can say that the PSNR decreases as the block size increases and RMSE increases as the block size increases and the compression ratio first increase to its maximum value and then starts decreasing as the block size is increased.Here,we are passing complete compressed array after removing zeros from the end and the image is coloured.

Conclusion

In conclusion, JPEG compression is a widely used method for compressing digital images that allows for efficient storage and transmission of images with minimal loss of quality. The JPEG compression process involves several stages, including color space conversion, discrete cosine transform (DCT), quantization.

In this Project, We have learned how a image is compressed using JPEG compression and how it is decompressed.The Various steps of doing that we learned through this project are:-

Encode/Compression:-

- First , we read the input image and divide it in the blocks of a chosen size
- In the next step, we subtract 128 from each pixel values
- In the next step ,we perform DCT on every block
- In the next step, we quantize each block by dividing it with a standard quantization matrix and then normalize the matrix by taking the round off every pixel value.
- In the next step, we convert the 2D matrix block in a single array by iterating over it in a zigzag manner. After this step we observe the many values at the end of this array becomes zero and we remove all these zeroes and add an symbol to denote the end of the array and finally we merge all these arrays to get a compressed array

Decode/Decompression:-

- First we divide the single compressed array into various small arrays by distinguishing it by the symbol that we used in decode to mark the end of arrays. Each one of these small arrays contains data of a block which will be merged to get the final image.
- In the next step, We append the zeros at the end to make the array size equal to the blocksize*blocksize.
- In the next step ,we perform reverse zigzagging to get the 2D matrix block from this small array
- In the next step, we multiply each of these blocks by Quantization Matrix .
- In the next step, we perform inverse DCT on each of these blocks.
- In the next step, we add 128 at each pixels of these blocks
- In the final step, we merge all these blocks to get the decompressed final image