JPEG- COMPUTING EXERCISE

ID: 20493425

DEGREE: MSC ELECTRONIC COMMUNICATIONS AND COMPUTER ENGNEERING

MODULE: DIGITAL COMMUNICATIONS EEEE3007

Table of Contents

Background on JPEG Image Compression

Image compression is a fundamental aspect of digital image processing, allowing efficient storage and transmission of images without compromising their quality. JPEG (Joint Photographic Experts Group) is one of the most common image compression techniques that uses lossy compression to reduce the size of image files without degrading the image quality past a certain threshold. This article provides an in-depth understanding of JPEG image compression, its effectiveness, and the results obtained through a MATLAB implementation. [1]

The two categories of image compression are lossy and lossless.

Lossy compression reduces the size of an image file by discarding some of the data, leading to a slight loss in image quality. This technique can reduce the file size without an obvious impact as it discards information in the image that is not visually significant meaning removing data that is difficult for the human eye to perceive. JPEG is a popular example of a lossy compression format.

a. Color Space Conversion: JPEG uses the YCbCr color space. The original image, in RGB format, is converted to YCbCr, which separates the luminance (Y) from the chrominance (Cb and Cr) information. This is done because the human eye is more sensitive to luminance than chrominance, allowing for more aggressive compression of chrominance.[2]

b. Downsampling: Chrominance channels are downsampled to reduce their resolution, as the human eye is less sensitive to color detail. Downsampling reduces the number of pixels in these channels without significant visual impact. [2]

c. Block Splitting: The image is divided into 8x8 pixel blocks to be processed.

d. Discrete Cosine Transform (DCT): Each 8x8 block is transformed from the spatial domain to the frequency domain using DCT. This process focuses most of the energy in an image into a few low-frequency coefficients, while the high-frequency coefficients contain less information. [3]

e. Quantization: The transformed coefficients are divided by a quantization matrix, which in this article has been predetermined in the file M.mat. The coefficients are then rounded off to integers. This process is lossy, as rounding off eliminates some data.

f. Zigzag Scan: The quantized coefficients use a zigzag scan, which groups low-frequency coefficients towards the beginning of the data stream. [4]

g. Entropy Coding: The zigzag scanned data is compressed using entropy coding techniques such as run-length encoding (RLE) and Huffman coding, which replace occurring patterns with shorter codes. [5]

Lossless compression, preserves all the data within an image, ensuring that the compressed image is identical to the original. Lossless compression techniques, such as PNG and GIF, are suitable for images with sharp transitions, text, or computer-generated graphics, where maintaining the original quality is necessary.

a. Predictive Coding: The image is scanned successively, and each pixel is predicted based on the value next to it. The prediction error is then computed for each pixel.

b. Entropy Coding: The prediction errors are compressed using entropy coding techniques such as Huffman coding or arithmetic coding. [5]

When decompressing a JPEG image, the process is reversed. For lossy compression, some information is permanently lost during compression, and the decompressed image will not be identical to the original. Lossless compression retains all information and allows the original image to be reconstructed precisely.[6]

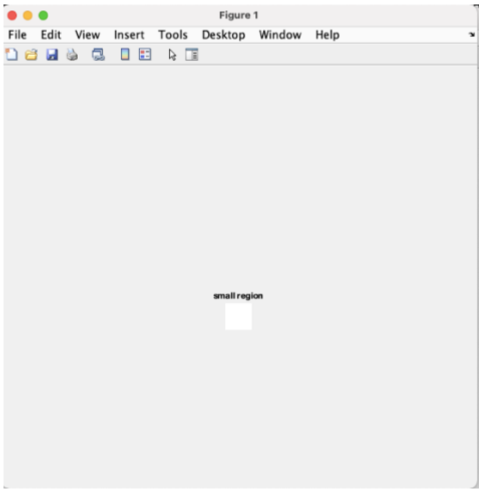
The results of quantization and reconstruction of the image produced:

PART A

The code selects an 8x8 pixel region from the Lena image starting at row 100 and column 200 and performs a 2D DCT operation using the dct2 function. It then calculates the Inverse DCT (IDCT) to restore the pixel values. This part demonstrates the DCT's ability to concentrate the image's energy into a few low-frequency coefficients, while the IDCT operation shows that the original pixel values can be perfectly restored. This shows the compression and decompression process for a small block/region of the image.

The results are shown below:

Figure 1 showing the 8x8 region for the row and column



The distribution of the DCT coefficient values and their positions in the matrix compared to the original 8 by 8 pixel values provide valuable information about the frequency content of the image.

Figure 2 showing the 8x8 DCT coefficient values and their positions in the matrix

```
8x8 DCT Coefficients:
   1.0e+03 *

    1.1479    0.0017   -0.0032   -0.0064   -0.0026    0.0025   -0.0021   -0.0029
   -0.0029   -0.0044   -0.0062   -0.0074    0.0050    0.0081   -0.0021   -0.0034
    0.0101    0.0031   -0.0001    0.0038   -0.0026   -0.0070    0.0009    0.0015
   -0.0042    0.0031    0.0091   -0.0017   -0.0096    0.0013    0.0037    0.0010
   -0.0049   -0.0026   -0.0017   -0.0146    0.0031    0.0089    0.0035   -0.0010
   -0.0016    0.0004   -0.0093    0.0071    0.0062    0.0013   -0.0057    0.0048
    0.0025    0.0002   -0.0004   -0.0002    0.0045   -0.0025   -0.0059    0.0017
   -0.0013   -0.0026   -0.0056   -0.0022   -0.0038   -0.0018   -0.0043    0.0008
```

Figure 3 showing the Original 8x8 Pixel values

```
8x8 Pixel Region:
   139   142   148   150   141   139   145   143
   144   146   146   145   138   148   143   146
   138   144   138   152   153   141   145   141
   141   133   142   153   141   137   142   142
   135   141   149   144   129   139   144   145
   149   150   143   136   140   144   144   143
   145   149   143   138   150   149   144   136
   145   149   148   145   146   145   142   148
```

Figure 3 shows the 8x8 Pixel matrix values. In an image, the black and white pixel values range from 0 to 255 respectively indicating that 0 is totally black and 255 is totally white. This means that 256 different shades of gray represent an image.

The DCT coefficients are calculated by transforming the 8 by 8 pixel region from the spatial domain to the frequency domain. The DCT coefficients represent the frequencies of the image content, with lower frequency components located in the upper-left corner of the matrix and higher frequency components located in the lower-right corner of the matrix.
The distribution of the DCT coefficients applies a “power-law” distribution, with a few large coefficients dominating the lower frequency components and many small coefficients contributing to the higher frequency components. This is the "energy compaction" property of the DCT, which means that most of the image energy is concentrated in a small number of coefficients.

The distribution of the DCT coefficient values in the matrix is important for image compression because it allows the higher frequency components, which contain fine details and textures, to be quantized or discarded without significantly affecting the overall quality of the image. This results in significant data compression, while preserving the visual quality of the image. This DCT process transforms signal to numerical data as seen in Figure 2 so it can be quantified and used for compression below.

In comparison, the original 8 by 8 pixel values in the spatial domain do not give any frequency information. The values represent the luminance or color values of the individual pixels in the region. Therefore, the DCT transformation is an important step in JPEG compression, as it allows the image to be represented more efficiently in terms of its frequency content, while discarding the higher frequency components that are less important for image perception.

Figure 4 showing the 8x8 IDCT coefficient values and their positions in the matrix

```
8x8 Inverse DCT:
  139.0000  142.0000  148.0000  150.0000  141.0000  139.0000  145.0000  143.0000
  144.0000  146.0000  146.0000  145.0000  138.0000  148.0000  143.0000  146.0000
  138.0000  144.0000  138.0000  152.0000  153.0000  141.0000  145.0000  141.0000
  141.0000  133.0000  142.0000  153.0000  141.0000  137.0000  142.0000  142.0000
  135.0000  141.0000  149.0000  144.0000  129.0000  139.0000  144.0000  145.0000
  149.0000  150.0000  143.0000  136.0000  140.0000  144.0000  144.0000  143.0000
  145.0000  149.0000  143.0000  138.0000  150.0000  149.0000  144.0000  136.0000
  145.0000  149.0000  148.0000  145.0000  146.0000  145.0000  142.0000  148.0000
```

After comparing the original pixel values with the IDCT values I observed that the IDCT output is extremely close to the original pixel values. This shows the efficiency of the IDCT operation in reconstructing the original spatial domain representation from the frequency domain representation provided by the DCT coefficients. The individual pixel values shows only minor discrepancies between the original and IDCT 8x8 regions. These differences are because of the precision of floating-point arithmetic in the calculations. The IDCT quality is visually indistinguishable from the original.

All this implies that DCT operation is lossless and is useful in JPEG for compactly representing the image data, by concentrating the energy in fewer coefficients, making it more suitable for compression.

PART B

The code quantizes the DCT coefficients using the provided quantization matrix (M) and attempts to restore the pixel values using inverse operations. This part demonstrates how the quantization process leads to a loss of information, as the restored pixel values are no longer identical to the original values. It highlights the trade-off between compression efficiency and image quality inherent in JPEG compression.

Figure 5 showing the Original 8x8 Pixel values

Original 8x8 Pixel Values:

139	142	148	150	141	139	145	143
144	146	146	145	138	148	143	146
138	144	138	152	153	141	145	141
141	133	142	153	141	137	142	142
135	141	149	144	129	139	144	145
149	150	143	136	140	144	144	143
145	149	143	138	150	149	144	136
145	149	148	145	146	145	142	148

Figure 6 showing the Restored 8x8 Pixel values

Restored 8x8 Pixel Values:

146.2865	146.2865	146.2865	146.2865	146.2865	146.2865	146.2865	146.2865
144.9471	144.9471	144.9471	144.9471	144.9471	144.9471	144.9471	144.9471
143.0529	143.0529	143.0529	143.0529	143.0529	143.0529	143.0529	143.0529
141.7135	141.7135	141.7135	141.7135	141.7135	141.7135	141.7135	141.7135
141.7135	141.7135	141.7135	141.7135	141.7135	141.7135	141.7135	141.7135
143.0529	143.0529	143.0529	143.0529	143.0529	143.0529	143.0529	143.0529
144.9471	144.9471	144.9471	144.9471	144.9471	144.9471	144.9471	144.9471
146.2865	146.2865	146.2865	146.2865	146.2865	146.2865	146.2865	146.2865

When comparing the restored pixel values with the original values, they values are close but there are some differences between the two. These differences are from the quantization step in the JPEG compression algorithm, which is responsible for the data reduction. This similarity shows that the restoration process is effective in recovering the spatial domain representation of the image from the quantized DCT coefficients. During quantization, the DCT coefficients are divided by a quantization matrix and then rounded to integer values, causing some loss of information, especially in the higher frequency components of the image.

Despite this, the visual appearance of the restored image is very similar to that of the original image, with most of the visual significant information preserved. This is because the quantization matrix used in the JPEG algorithm prioritizes the retention of lower frequency components, which carry more visually important information, while discarding or reducing the precision of less significant high-frequency components as explained above.

The restoration process caused some loss in image quality, but also achieved significant data reduction. The data reduction gotten usually depends on the level of quantization applied. The higher the quantization step, the greater the data reduction, but at the cost of lower image quality. It seems that a balance is found between the desired level of compression and the acceptable loss of image quality for a given image.
The dequantization of coefficients and the application of the IDCT, recovers the pixel values from the compressed data.

PART C

The code applies the JPEG compression process (DCT, quantization, dequantization, and inverse DCT)to the entire Lena image by using the blockproc function. It then calculates the mean squared error (MSE) and peak signal-to-noise ratio (PSNR) to quantitatively assess the restored image's quality. By altering the quantization matrix's scaling factor, the code explores the relationship between image quality and compression ratio, generating a plot to visualize the trade-off.

Figure 7 showing the Restored Lena image

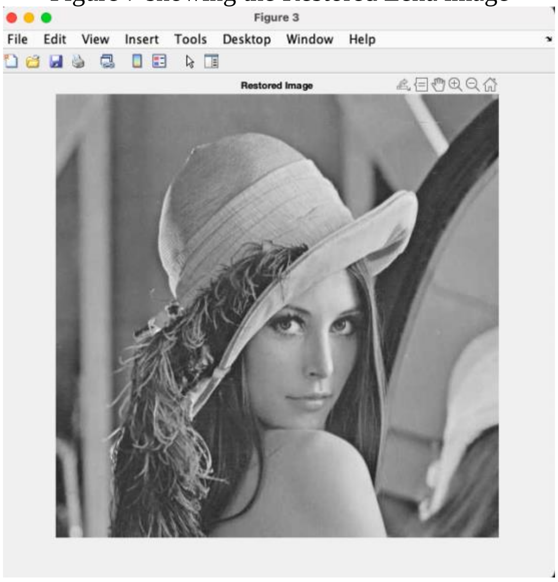
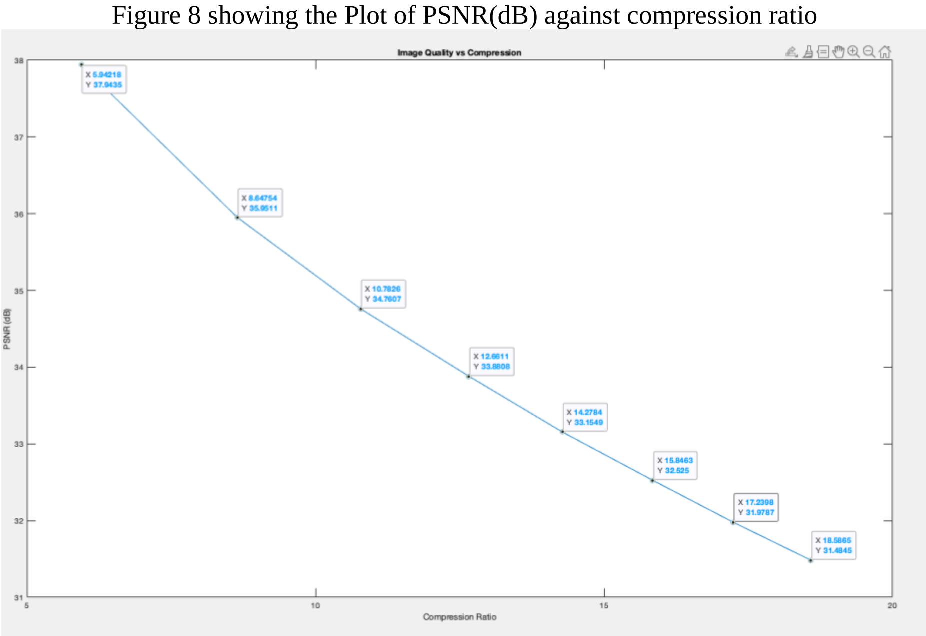


Figure 8 showing the Plot of PSNR(dB) against compression ratio



The process from Part B was repeated for the entire Lena image using the blockproc function. The restored image showed noticeable degradation in visual quality when compared to the original Lena image, in the high-frequency areas, such as edges and textures. The restored image appears smoother than the original Lena image. However, the overall structure and content of the image were still recognizable. The quantitative measure of image quality the Peak Signal-to-Noise Ratio (PSNR) was calculated using the mean squared error (MSE) between the original and restored image. The MSE is the average of the squared differences between the corresponding pixel values in the original and restored images. This formula psnr = 10 * log10(255^2 / mse); was used with 255 being the maximum possible pixel value of the image. The compression ratio was computed to estimate the amount of data reduction achieved. The M matrix was scaled, and the process was repeated to obtain different compression ratios.

A graph of PSNR against the compression ratio was plotted in Figure 8 above showing a trade-off between image quality and compression. This shows the balance between the desired image quality and the amount of data reduction.

The graph above shows the relationship between the compression ratio and PSNR (peak signal-to-noise ratio) values for the image, with PSNR values ranging from 31.5 to 37.9 dB which is good as it is above 30dB as higher is better. This graph helps us understand how the image quality degrades as the compression ratio increases.
From Figure 8, it is observed that as the compression ratio increases, the PSNR value decreases. This is because a higher compression ratio means more data is being discarded, which gives a lower image quality.
The quality of degradation in the graph above is not linear with the increase in the compression ratio. At the lower compression ratios, the PSNR values decrease more gradually. Meaning that the image quality is not significantly affected when the compression ratio is low. However, as the compression ratio increases, the PSNR values drop more rapidly, indicating a more noticeable loss of image quality. This is because more substantial amounts of data are being discarded from the image, therefore a greater loss in visual quality.

Analysis Conducted and the Quality of the reconstructed images and the amount of data reduction achieved

The reconstructed image has a degradation in visual quality, particularly in high-frequency areas such as edges and textures. However, the overall structure and content of the image are still recognizable. This shows that JPEG compression can achieve data reduction while preserving the main features of the original image.

For the analysis of Part A, an 8x8 region was selected from the Lena image, and 2D DCT is applied to the small region. The DCT coefficients are then printed, followed by the inverse 2D DCT. The distribution of the DCT coefficient values shows that most of the energy is concentrated in the top-left corner of the matrix (low-frequency components), which is a common observation for natural images. The inverse 2D DCT result is almost identical to the original 8x8 pixel values, showing the lossless nature of the DCT transform.
In Part B, the same 8x8 region is selected, and the DCT coefficients are quantized using the 'M' matrix provided. The quantized coefficients are then dequantized, and an inverse 2D DCT is applied to restore the pixel values. The restored pixel values have some differences from the original values due to the quantization step, which introduces some loss in the image data. This loss is a trade-off for achieving data reduction (compression).
In Part C, the entire Lena image is processed in 8x8 blocks, and the same steps from Part B are applied to each block. When the restored image is compared visually to the original Lena image, there is some noticeable degradation in quality, especially in areas with high-frequency details (e.g., sharp edges or textures). To quantitatively assess the image quality, the mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are calculated. Lower MSE and higher PSNR values indicate better image quality. [7]

A higher PSNR value means better image quality, whilst a higher compression ratio means more data reduction. From the graph plot in Figure 8, at PSNR = 31.5 dB at Compression Ratio 18.6, the reconstructed image has a PSNR value of 31.5 dB, which is just above the 30 dB threshold that is often considered acceptable for image quality.

The data reduction achieved is significant, as it compressed the image to 5.4% of its original size. This value was gotten by following this calculation:
Percentage of data reduction = (1 / Compression Ratio) * 100
Percentage of data reduction = (1 / 18.6) * 100 = 5.4%

When PSNR is 32 dB at Compression Ratio 17.2, the PSNR value is slightly higher at 32 dB, indicating slightly better image quality than the previous compression ratio. The data reduction achieved is still substantial, compressing the image to 5.8% of its original size. This has size of data reduction means there are minor distortions.

When PSNR is 33 dB at Compression Ratio 14.3, the PSNR value of 33 dB means the quality of the image is good and the compression ratio of 14.3 results in the image being compressed to about 7.0% of its original size. This level of compression provides a good balance between image quality and data reduction.

For PSNR = 35.9 dB at Compression Ratio = 8.6 the PSNR value is 35.9 dB, which suggests excellent image quality. The image is compressed to about 11.6% of its original size, which is still a significant data reduction. This level of compression provides a good balance between image quality and data reduction for applications requiring higher image quality.

When PSNR = 37.9 dB at Compression Ratio = 5.9 which is the lowest compression ratio, the PSNR value is 37.9 dB, indicating very high image quality. The image is compressed to about 16.9% of its original size, which is a lower data reduction than the other compression ratios. This quality is outstanding compared to the above and is suitable for applications that prioritize image quality over data reduction.

For some applications, a high compression ratio and significant data reduction might be vital, such as when transmitting images over limited bandwidth networks or when storage space is constrained. In these situations, the reduced image quality is an acceptable compromise for the benefits of reduced file size and faster transmission.
On the other hand, for applications that demand high image quality, such as professional photography or medical imaging, the reduced image quality at a compression ratio of 18.5 might not be acceptable. In these cases, a lower compression ratio that preserves more of the image quality would be more appropriate.

Conclusion

In conclusion, the MATLAB implementation of the JPEG image compression technique demonstrates the trade-off between compression ratio and image quality. The provided code effectively compresses the Lena image through DCT, quantization, dequantization, and inverse DCT. The experiment highlights the impact of various quantization levels on the resulting image quality and compression ratio.

In Part A, it was observed that the DCT operation is lossless and concentrates most of the image's energy into a few low-frequency coefficients, making it suitable for compression. Part B showed how quantization introduces some loss of information but still preserves visually significant information in the image. The restored pixel values are close to the original values, indicating that the process is effective in recovering the spatial domain representation of the image. Part C applied the compression process to the entire Lena image, and by altering the quantization matrix's scaling factor, the code explored the relationship between image quality and compression ratio.

The graph in Figure 8 clearly shows the trade-off between compression ratio and image quality (PSNR). As the compression ratio increases, the image quality decreases, which is expected due to the lossy nature of JPEG compression. However, at lower compression ratios, the PSNR values decrease more gradually, suggesting that image quality is not significantly affected when the compression ratio is low.

This article shows how a balance there is a balance between the desired level of compression and the acceptable loss of image quality for a given image. For example, at the compression ratio of 14.3, the PSNR value is 33 dB, which indicates good image quality, while the image is compressed to approximately 7.0% of its original size. This level of compression provides a good balance between image quality and data reduction.

This experiment is very educating and highlights the effectiveness of JPEG image compression in attaining significant data reduction while preserving the main features of the original image. MATLAB was the coding app used to get the results and the tools and functions in it helped understand and explore the fundamental aspects of JPEG compression and the trade-offs involved in this technique.

References

[1] Meghanathan, N., Chaki, N. and Nagamalai, D. (2012) *Advances in computer science and information technology. Computer Science and Information Technology Second International Conference, CCSIT 2012, Bangalore, India, January 2-4, 2012. proceedings, part III*. Berlin, Heidelberg: Springer Berlin Heidelberg.

[2] *Chroma subsampling* (2023) *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Chroma_subsampling (Accessed: 03 May 2023).

[3] (No date) *The discrete cosine transform (DCT)*. Available at: https://users.cs.cf.ac.uk/dave/Multimedia/node231.html (Accessed: 03 May 2023).

[4] (No date a) *Zig-Zag Scan*. Available at: https://users.cs.cf.ac.uk/dave/Multimedia/node238.html (Accessed: 03 May 2023).

[5] *Entropy coding* (2023) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Entropy_coding (Accessed: May 3, 2023).

[6] Awati, R. (2021) *What are lossless and lossy compression?*, *WhatIs.com*. TechTarget. Available at: https://www.techtarget.com/whatis/definition/lossless-and-lossy-compression (Accessed: May 3, 2023).

[7] Haweel, A.H. (2018) *Development of Hiding a Secret Message in an Image by Encryption Using Fuzzy Logic*. thesis.