Demet Yayla - 2019400105
Kadir Ersoy - 2018400252

Image Compression

We have chosen to implement two image compression methods from literature.

- Discrete Cosine Transform Compression
- Wavelet Transform Compression

Discrete Cosine Transform

DCT compression method takes its foundation from the fact that in general, majority of the significant information of an image is condensed in just a few coefficients of the DCT transformed version, which property also referred to as “Energy Compaction”. Thus, DCT finds wide range of utilization in image and video compression. Typically, low-frequencies in cosine transformed version of the image reflects the overall structure and global variations while high-frequencies reflect fine details and local variations.

Implementation Details

I observed some differing implementations. Some of the algorithms contained an additional step called encoding while compressing. I skipped that part since it wasn’t the significant operation in the algorithm.

The algorithm basically divides the image into pixel blocks that are NxM sized blocks. This block analysis enables us to have a more homogeneously and smoothly implemented transform. We in parallel define a quantization matrix later to use while truncating the block. Per each block, the algorithm first takes the discrete cosine transform. This transformation finds a utilization in many other image processing and compression algorithms (like JPEG image compression). It basically separates image into different frequency components. A low value in the transformed coefficients typically implies a less significant frequency.

The various algorithms have differing approaches at the reduction and damping of these less significant frequencies. Some just dim (make zero) the values less than a certain threshold value; some choose more specialized matrices for division. I preferred a mixture of these. To create a more basic method, I simply created a matrix filled with the same quantization coefficient. Then, I did element-wise division with this quantization matrix, operation done per each color channel (R-G-B).

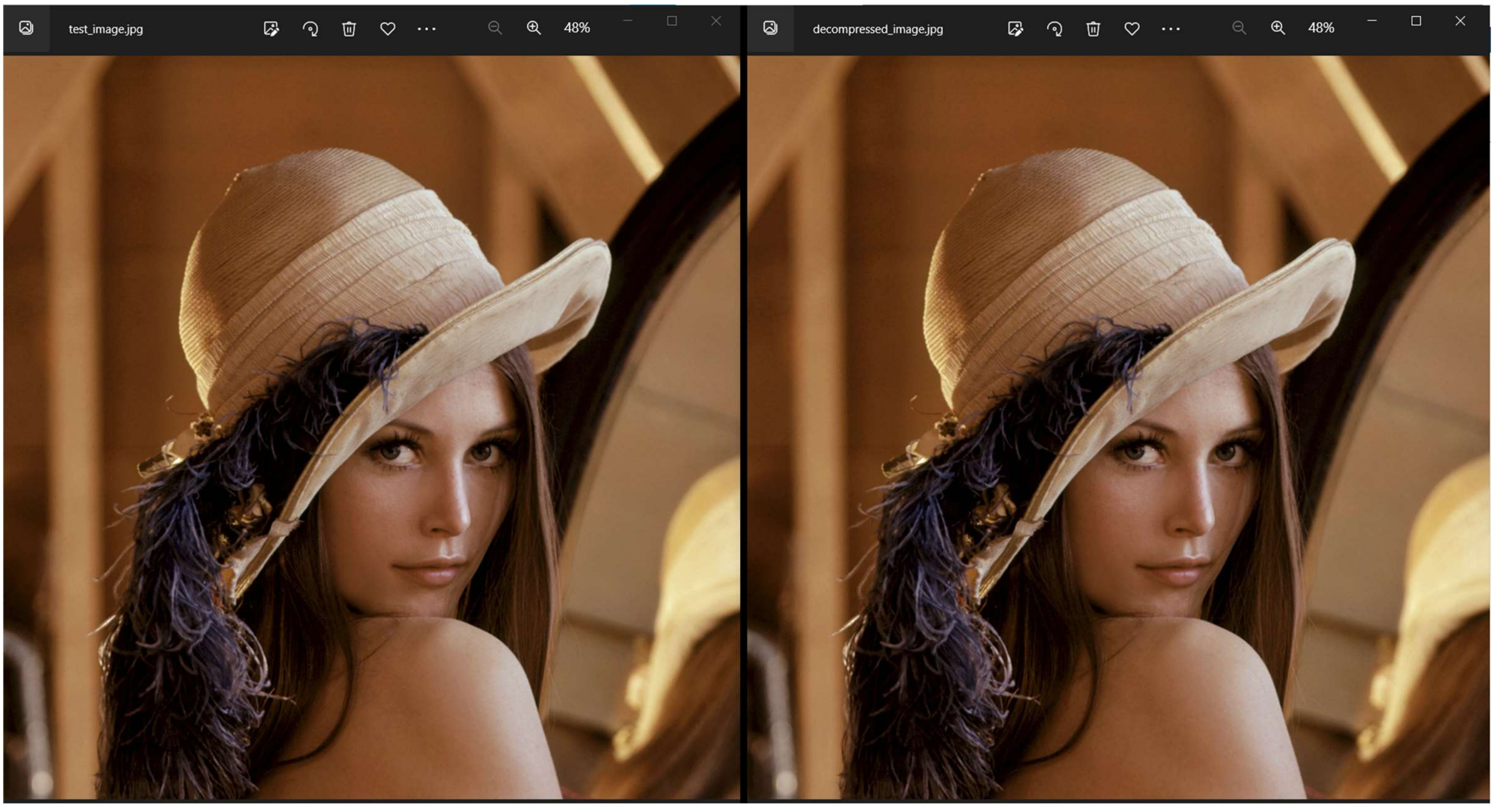
Theoretical analysis of this division is as follows: If I only dimmed the frequencies less then a certain value, there would have been less information loss. Now, even along with the ones I caused to become zero, other values frequency values lost some precision up to a level (since I chose coefficient to be 64, the last 6 digits of twos representation of the frequency values). This has no apparent loss in my test case.

Another comparison caught my attention: in homework 2, we implemented a basic recovery of damaged images by hiding a copy of the image inside the image. But we had to loose precision because we couldn’t keep all the digits due to spatial restrictions. When recovered, the resolution dropped due to this information loss. The same amount of digit shift isn’t observed in our case when we do this shift on the discrete cosine transformed image.

Observation from My Test of the Code I Wrote

Numeric data:

Original size of the image:	722 KB
Compressed size of the image:	111 KB
Decompressed size of the image:	267 KB



From left to right: decompressed image, original image

Demet Yayla - 2019400105
Kadir Ersoy - 2018400252

Wavelet Transform

The wavelet transform breaks down a signal into a set of wavelet functions at different scales, allowing for a more detailed analysis of its frequency content and temporal localization.

Unlike the Fourier transform, which represents a signal as a sum of sinusoidal functions of different frequencies, the wavelet transform uses wavelet functions that are more localized in both time and frequency.

The wavelet transform operates by convolving the signal with a set of wavelet functions, known as the mother wavelet, which are scaled and translated versions of a basic prototype wavelet. By varying the scale and position of the wavelet function, the transform can extract different features from the signal. The resulting wavelet coefficients represent the contribution of each wavelet function to the signal at different scales and positions.

Idea behind the wavelet compression is to decompose the image into different frequency components using the wavelet transform.

The compression process

- **Wavelet Decomposition**: The input signal or image is decomposed into a set of wavelet coefficients using the wavelet transform. As described above, wavelet coefficients represent contribution of each wavelet and the waveletes are acquired by convolving the input data with a set of wavelet functions, which are scaled and translated versions of a basic prototype wavelet.
- **Quantization**: The wavelet coefficients are quantized. By representing the coefficients with less bits, some loss introduced but necessary size of data is reduced.
- **Encoding**: The quantized wavelet coefficients are encoded.
- **Reconstruction**: To reconstruct the compressed signal or image, the encoded data is decoded and inverse wavelet transform is applied. The inverse wavelet transform combines the wavelet coefficients to recreate the original signal or image.

Wavelet compression offers several advantages over DCT. It can effectively capture both localized and global features of a signal or image, leading to better preservation of important details. Additionally, wavelet compression can achieve higher compression ratios while maintaining visual quality, especially for images with sharp edges and textures.