CMPE 362 - Introduction to Signals for Computer Engineers
Homework 3

Fatih Alagöz, TA: Mete Tuluhan Akbulut

Grading

There is no submission. You will have randomized questions about this project during the final, and you will need MATLAB to solve questions. Make sure that you understood the techniques described here, you saved your code that solves the questions below and MATLAB is running on your computer before the exam.

The required images(peppers, fruits) can be found on Moodle.

Question 1 - Image Watermark with LSBs

In this homework, you will learn to watermark images with an image or text and extract the watermark.

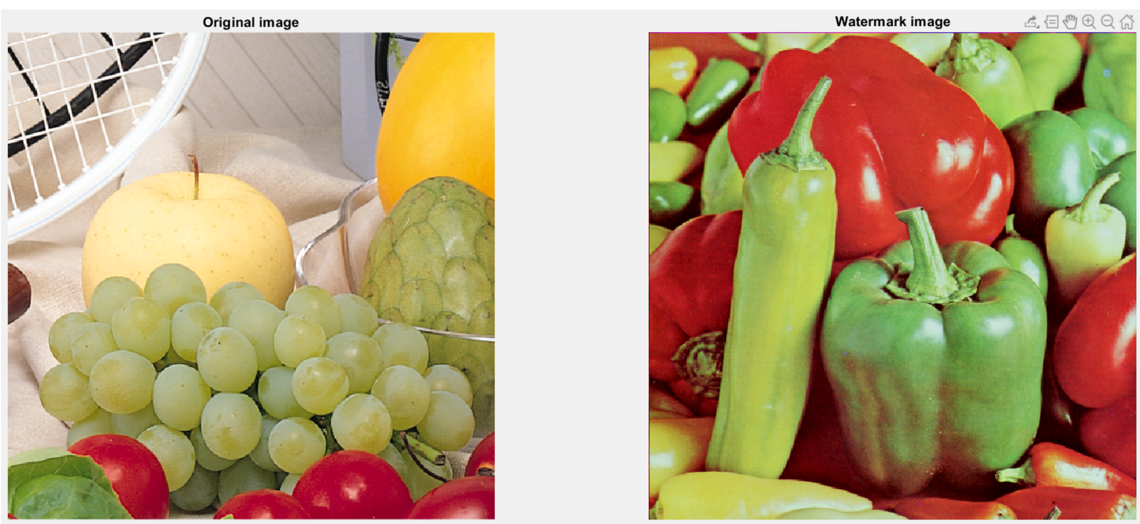

Figure 1: Initial images

The first method is the Least Significant Bits (LSBs) method. Remember that every pixel of an image has 3 color channels (red, green, and blue), which are represented by values from 0 to 255.

We can also use 8 bits to convert these values to unsigned binary numbers. The most significant bit (MSB) of the binary number is the left-most bit, and the least significant bit (LSB) is the right-most bit. If we change the least significant bits of an image, this operation may be visually unnoticeable (One bit change: ± 1, two bits change: ± 3, three bits change: ± 7). The trick is that we will write the most significant bits of the watermark image to the least significant bits of the original image so that we can transfer the watermark image without distorting the original image. See the figure 2 for illustration.

Two images (peppers.png and fruits.png) can be found on Moodle. You will embed the peppers image into the fruits image with the described method above. The shapes of the 2 matrices are the same so that you can embed the image pixel by pixel. Here, you should check the procedure w.r.t. the number bits used (Figure 3 and 4). Calculate the 2-norm of differences matrices (fruits image $-$ watermarked fruits image) for 3 channels and find the average (For 3 LSBs: 540.2155, For 6 LSBs: 3.8785e+03).

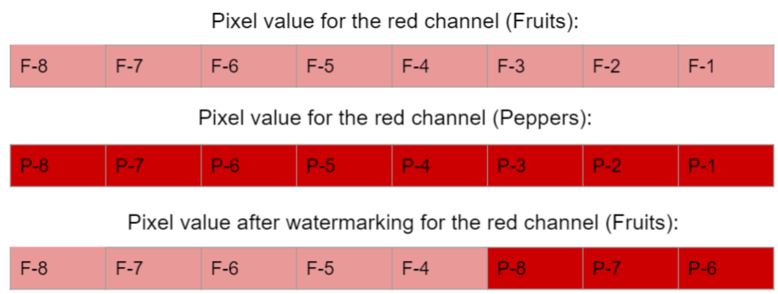

Figure 2: Watermarking with LSBs. The numbers show significance (8: MSB)

The second part of the question is extracting the watermark. Assume that only the watermarked fruits image and the number of LSBs used for embedding are given. You should be able to extract the watermark image by reversing the method above.

Question 2 - Text Watermark with LSBs

In this part, we will follow the same procedure to embed text into the fruits image. Assume a text file is given to you. Read this text file in MATLAB. Then, find the binary representation of the ASCII codes of each character and place them into a matrix. For the text "I did not receive any help while doing this homework. All effort is my own.", you should obtain a matrix with 75 rows and 7 columns. Figure 5 shows first 15 rows for the given text. Place these bits to the LSB of the first 75 rows and 7 columns of the red channel of the fruits image. The resulting image will look like figure 6 if I use the most significant bit.

The second part of the question is extracting the watermark. Assume that only the watermarked fruits image, the color channel, and the location of the watermark are given. You should be able to extract the text by reversing the method above. Write that text to a .txt file.

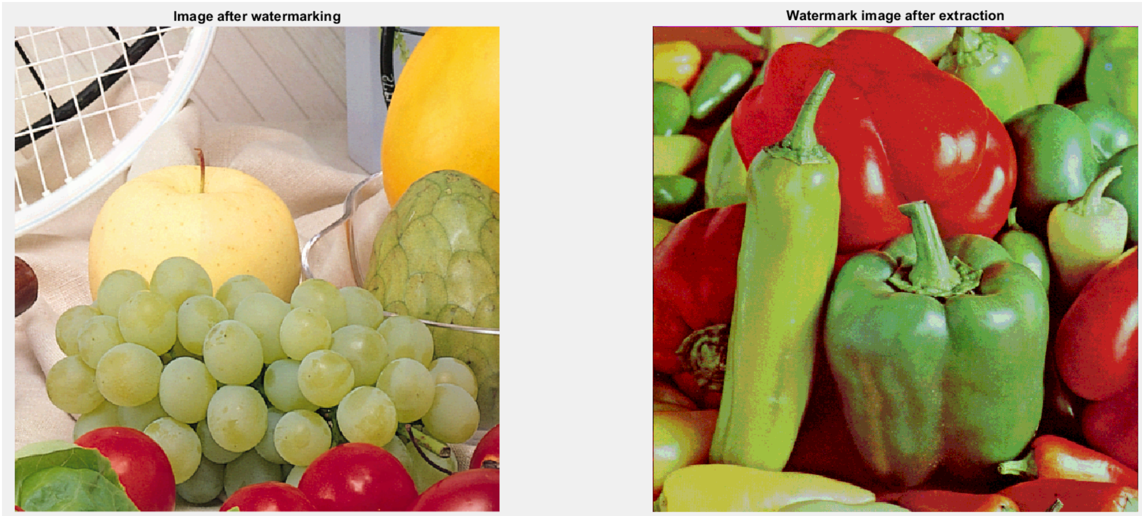

Figure 3: Watermarking with 3 LSBs

Question 3 - Image Watermark with Discrete Cosine Transform

The second method is the Discrete Cosine Transform. It is a method to represent an image as a sum of sinusoids of varying magnitudes and frequencies. Learn how to use this transform with the url https://www.mathworks.com/help/signal/ref/dct.html. This transform is also used in JPEG compression and there are many applications including different ways of watermarking. Assuming that you are comfortable with color channels, we will use grayscale images in this question. Convert fruits image and peppers image to grayscale.

DCT transform gives a matrix of the same size. The top-left of the resulting matrix has the lowest frequency and the frequency increases when you go right and down on the matrix. In addition, we assume that most of the energy of natural signals is stored in low-frequency coefficients. Consider DCT transforms of grayscale fruits and peppers images (colormap parula and colorbar are used for illustration.) in the figure 7. If we take a slice from the top-left of the DCT matrix (shown with red line) we can reconstruct the image without too much information loss. We will use this trick to embed the watermark. Figure 8 summarizes the procedure. We will take a diagonal slice from the DCT matrix of the watermark and place it to the lower left of the DCT matrix of the original image. The diagonal is the secondary diagonal of the top left 400×400 matrix. The resulting image without scaling and its DCT transform is illustrated in the figure 9. Notice how high-frequency terms affect the image. Therefore, we should scale the slice before placement to make our watermark unnoticeable. I selected 1/max value of the DCT matrix of the original grayscale fruits image. The watermarked image and the watermark after reconstruction are illustrated in figure 10. Calculate the 2-norm of the difference matrix (grayscale peppers image − grayscale peppers image after reconstruction, 409.7138).

Figure 4: Watermarking with 6 LSBs

75x7 double

	1	2	3	4	5	6	7
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	1	0	0	1	0	0
4	1	1	0	1	0	0	1
5	1	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	1	1	0	1	1	1	0
8	1	1	0	1	1	1	1
9	1	1	1	0	1	0	0
10	0	1	0	0	0	0	0
11	1	1	1	0	0	1	0
12	1	1	0	0	1	0	1
13	1	1	0	0	0	1	1
14	1	1	0	0	1	0	1
15	1	1	0	1	0	0	1

Figure 5: First 15 rows of the resulting matrix for the given text

Figure 6: Text watermark with the MSB

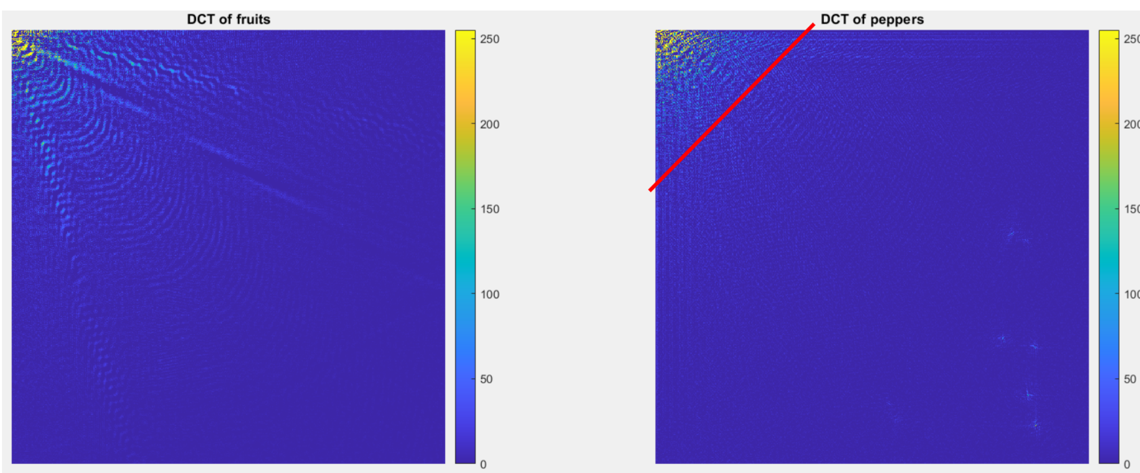

Figure 7: Discrete Cosine Transform Illustration

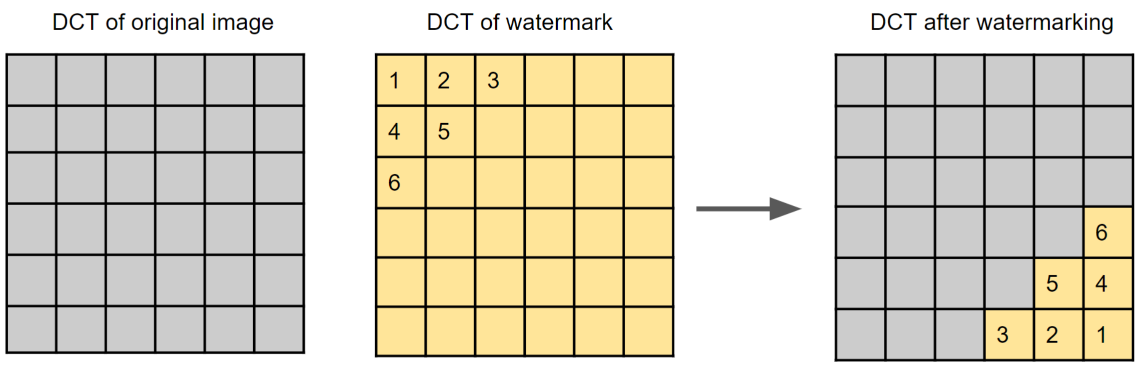

Figure 8: Procedure summary

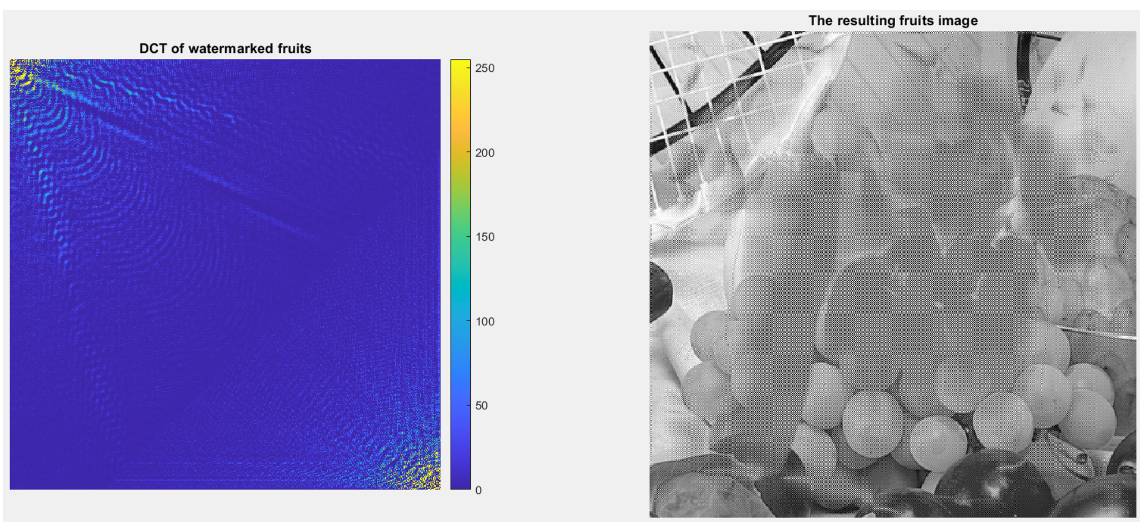

Figure 9: Procedure without scaling

Figure 10: Results with scaling