CS355 - Digital Forensics
Report for labs 1-3

Mikas Michelevičius
ID: 1835495

February 15, 2021

Contents

1 Lab 1

1.1 Exercise 1

Image of Lena is loaded using **imread** command. The image is stored in matrix of size (512, 512, 3) where each red, green and blue channel corresponds to one of three matrices. Thus, the components are retrieved such as **redChannel = I(:, :, 1);**. Using 'gray' colormap channels are displayed, see Fig 1.

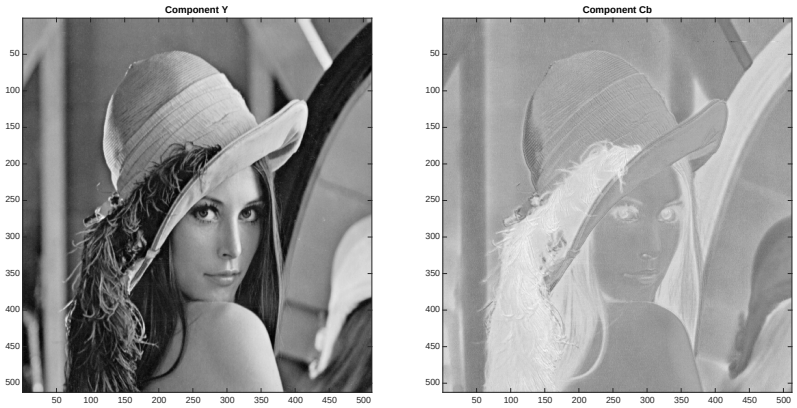
(a) Red Channel

(b) Green Channel

(c) Blue Channel

Figure 1: RGB Channels

The original image is then reshaped into a matrix of size (512*512,3) and multiplied by the scale factor matrix provided. Cb and Cr components are increased by 128 units and the matrix is reshaped back into original (512,512,3) size where Y, Cb and Cr component corresponds to one of three matrices. These components are retrieved such as **y = ycbcrImg(:,:,1);** and displayed, see Fig 2.



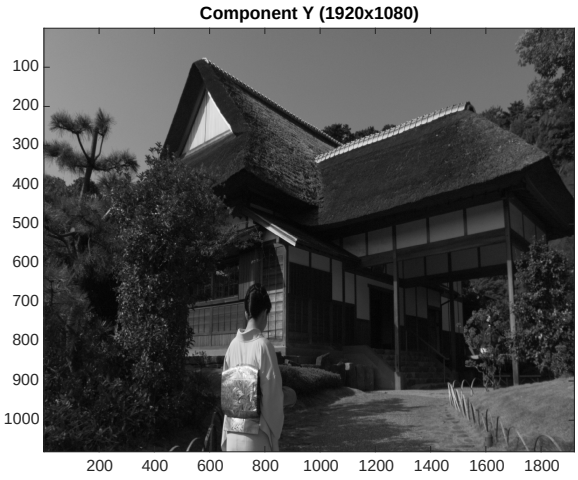
(a) Y Component (b) Cb Component

(c) Cr Component

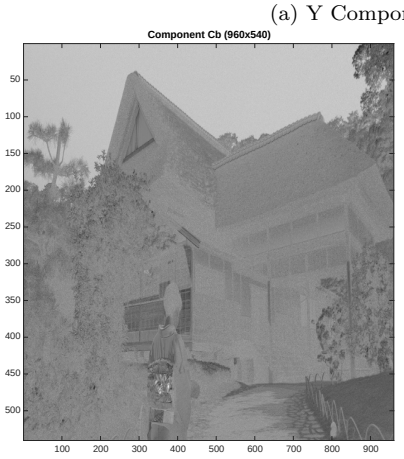
Figure 2: YCbCr Components

1.2 Exercise 2

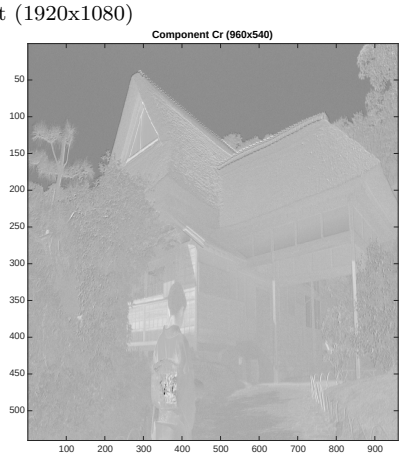
Knowing that the frame is in YCbCr 4:2:0 format the assumption is made that the dimensions of Cb and Cr components are (960,540). The vector of size (1920*1080+2*960*540) is read from file. Each component is initialized by their corresponding vector locations and the components are reshaped into the matrices which are displayed, see Fig 3.



(a) Y Component (1920x1080)



(b) Cb Component (960x540)



(c) Cr Component (960x540)

Figure 3: YCbCr Components

Four downsampled versions of Y component are calculated using **mat2cell**, **cell2mat** and **cellfun** functions. The downsampled components together with Cb component are then split into 8x8 blocks where each block of each downsampled component is compared to the Cb component's respective block. Based on the highest correlation of each block, corresponding suspect count is increased.

- Your task is to use digital forensics method to determine which suspect shot illegal video.
- Calculations resulted in suspect D having the highest number of correlated blocks, which means that suspect D is guilty. The other three suspects resulted in smaller correlation numbers, see Table 1

Suspect ID	No. of Blocks
D	3290
A	1985
C	1767
B	1118

Table 1: Correlated blocks of each suspect

2 Lab 2

2.1 Exercise 1

Function loads image to be enhanced and a reference image. Reference image is displayed together with its histogram. The input image to be enhanced is split into R, G and B channels where each of them will be used to perform histogram matching. Histograms of each channel are computed using **imhist** function, after normalization they are displayed, see Fig 4, first row.

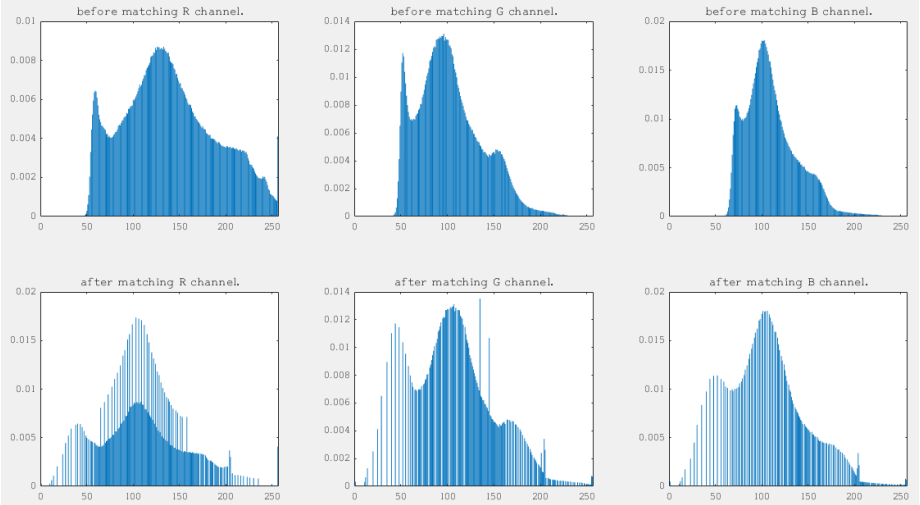


Figure 4: Histograms before and after histogram matching

On each of the channel histogram matching is performed with respect to the reference image histogram. The normalized histograms of each channel after matching are displayed, see Fig 4, second row. The matched R, G and B channels are then concatenated into a single image. Enhanced image is now displayed beside the initial image before enhancement, see Fig 5.

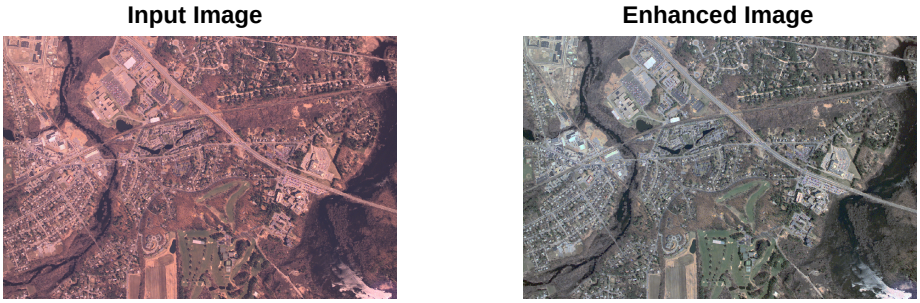


Figure 5: Original and enhanced images

2.2 Exercise 2

The function loads the image which contains repetitive (periodic) noise. The aim is to effectively suppress the noise by using a notch filter. As known, to suppress the periodic noise, notch filter has to be applied on the unwanted peaks from the image's Fourier transform. This exact thing is done next, where Fourier transform of the image is displayed and the unwanted peaks can be easily seen, where the unwanted peaks are the lines dividing the image into equal 16 squares. In two nested for cycles the filter of type given to the function as input is applied in the locations of unwanted peaks of image's Fourier. After that, the inverse of Fourier function is performed and resulting image is displayed. Trying all three filter types resulted in the 'ideal' filter resulting in the cleanest image together with D0 parameter as 0.05*PQ(1), see Fig 6.

Figure 6: Clean image

3 Lab 3

3.1 Exercise 1

Grayscale Lena image is loaded and using **bitget** function first two and last two bitplanes are extracted (1,2,7,8) and displayed, see Fig 7.

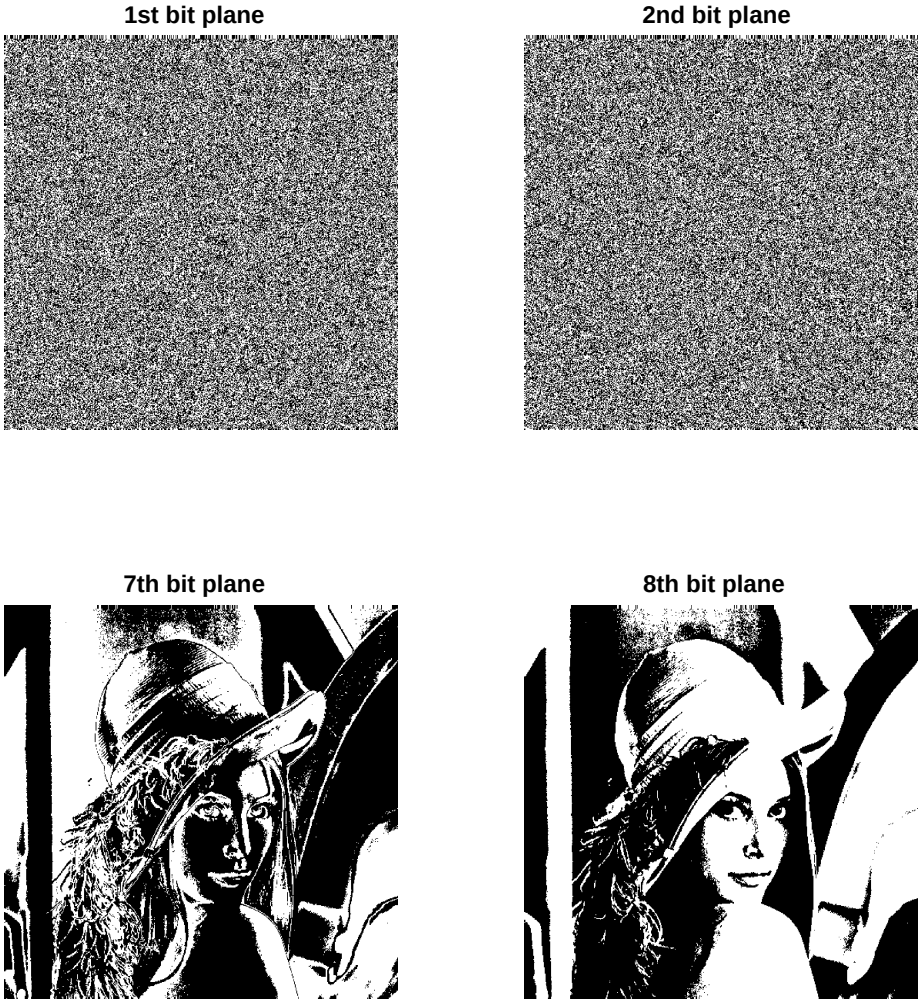


Figure 7: Bit planes

Then the watermark image is loaded. After performing several steps, the image is binarized. Firstly the appropriate threshold value is selected which is the value between the max and min values of the image. In this case, threshold value t is 182. All intensities of the image above t are represented as 1 and 0 otherwise using **imbinarize** function. Then, using **imcomplement** function

negative image is created. Both binarized and negative binarized images can be seen in Fig 8.

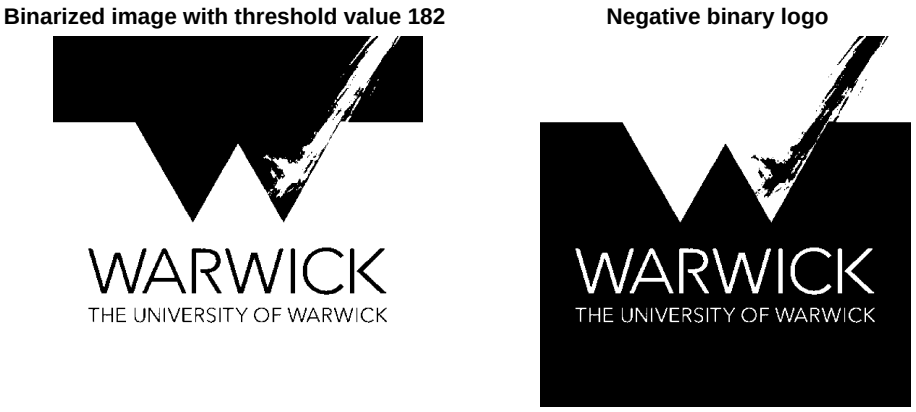


Figure 8: Logo and negative logo

Lena is then watermarked by replacing its least significant bit (LSB) with the binarized negative logo image. The SSIM value between original Lena image and the watermarked image results in value of 0.9979. After compressing image into JPEG format, the image is now loaded and watermarked logo from LSB plane is retrieved and displayed. The observations are clear that such watermarking technique is not robust since the watermark can be easily damaged even by performing image compression. The watermarked image and the watermark after image compression can be seen in Fig 9.

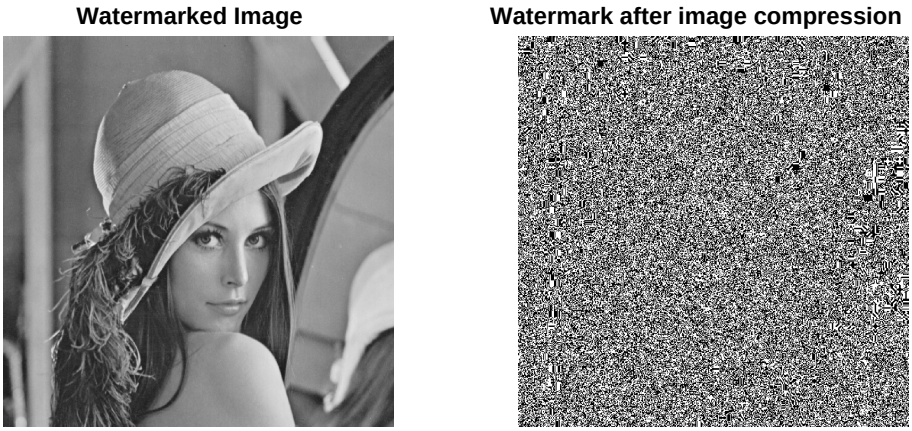


Figure 9: Watermarked lena and recovered logo

3.2 Exercise 2

A random watermark vector **w** of length n is computed using **randn** function and made to have zero mean and unit standard deviation. Lena image is loaded and converted to DCT coefficient matrix. Both original image and DCT matrix are displayed, see Fig 10.

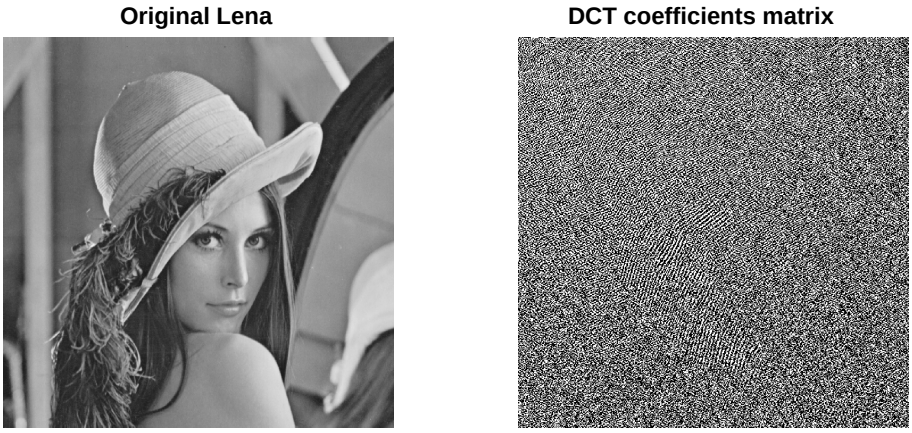


Figure 10: Original Lena and DCT matrix

n largest DCT coefficients of absolute value are found with their corresponding locations, excluding the first (largest) DCT coefficient. Watermarked coefficients are generated and placed in the DCT matrix in the right locations. Applying inverse DCT on the watermarked coefficients the watermarked images are created.

Figure 11: Images with N largest coefficients

The experiment is performed using different n values (100,1000,1500). However the changes in the images visually are not seen, see Fig 11. On the other hand, SSIM between original image and the watermarked images are computed, and the results are slightly different, see Table 2.

N coefficients	SSIM
100	0.9979
1000	0.9945
1500	0.9930

Table 2: SSIM with different values of N