Wrocław University of Science and Technology

Artificial Intelligence and Computer vision laboratory classes

Lab 4 – Quantization, thresholding, DFT

Kacper Harężga 249111

1. Introduction

During this labs we got familiar with the quantization, thresholding and Discrete Fourier Transforms of the image with use of OpenCV framework.

2. Code description

a. Used libraries

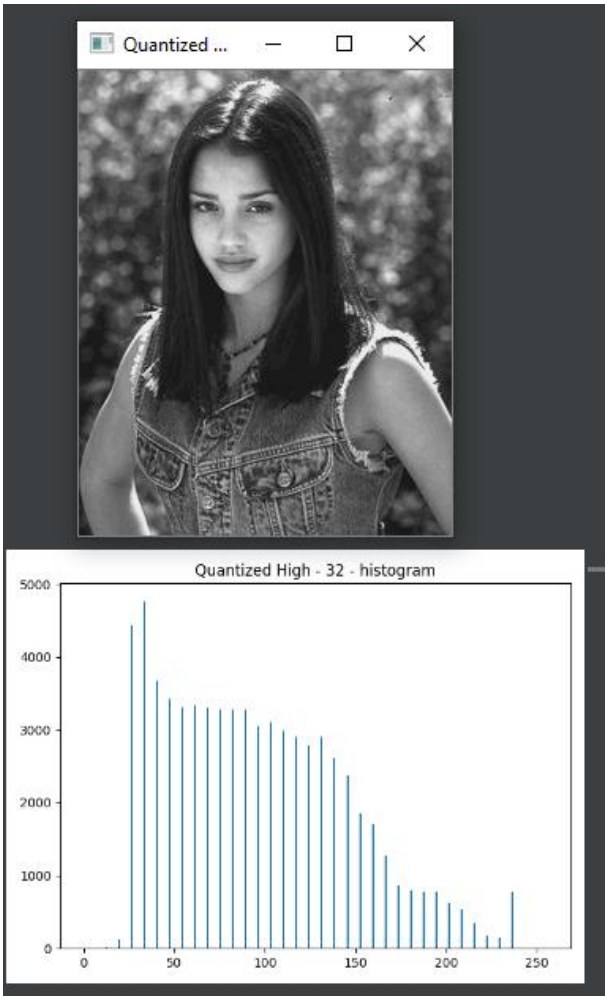
In the beginning of the provided source code naturally one can distinguish three essential libraries needed for the our realization. Short description of each lib:

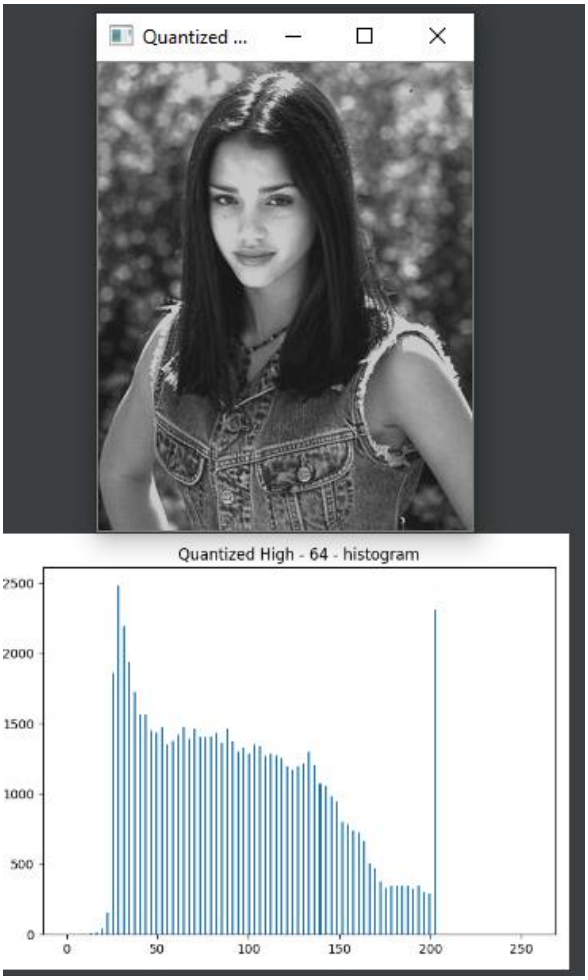
- **Matplotlib** imported as plot is used for the histogram plotting and titling plots.
- **OpenCV** the most crucial import. Needed for the resizing, interpolating and histogram calculation.

b. Code description

- **Task 1&2**

In order to realize first and second task I have implemented function **quantizator()** which performs quantization of the image with the given level. Right below results of the quantization are placed. The difference between level 32 and 64 is so small that I might be neglected by most peoples. The biggest difference is in the histogram plots, quantizied image contains bigger amount of pixels with values close to white or black

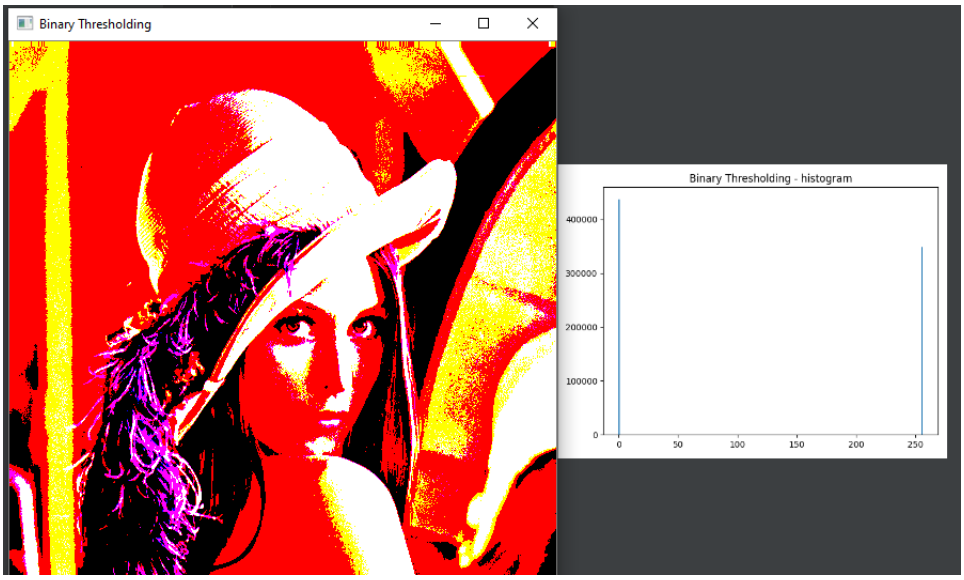

Quantized ...
Quantized High - 32 - histogram
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

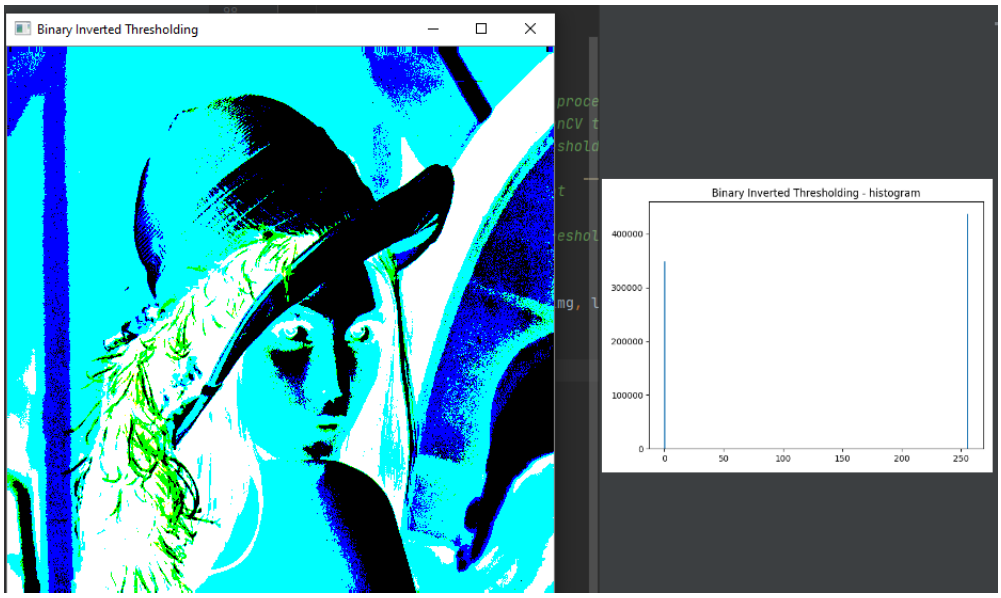

Quantized ...
Quantized High - 64 - histogram
2500
2000
1500
1000
500
0
0
50
100
150
200
250

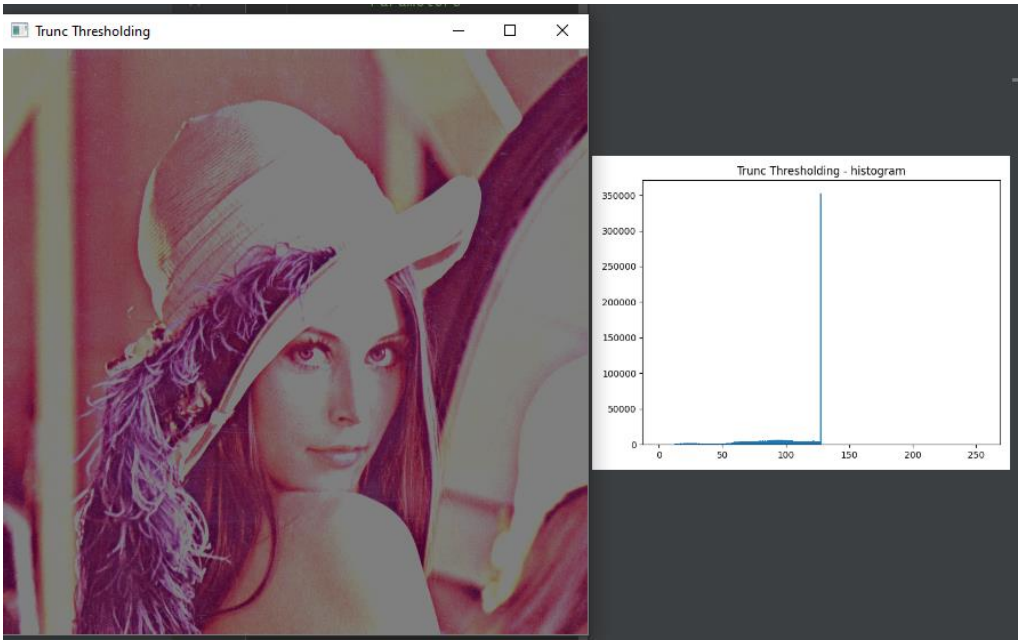
- **Task 3**

The goal of the third task was image thresholding. I have used OpenCV function combined in with histogram function. Whole description of the function was placed in the source code with respect to the rules of the code documentation. I have tested the following types of thresholding: **Binary, Binary Inverted and Trunc.** The result of the thresholding are placed right below.

```
def img_thresholding(img, lvl, type_of_thresh, title):
    """Performs tresholding of the image.

            Parameters
            ----------
            img : OpenCV type
                The image to be processed
            type_of_thresh: OpenCV type
                Type of the thresholding
            title : str
                Title of the plot
            lvl : int
                Level of the thresholding

            """
    ret, thresh = cv.threshold(img, lvl, 255, type_of_thresh)
    histogram(title, thresh)
```







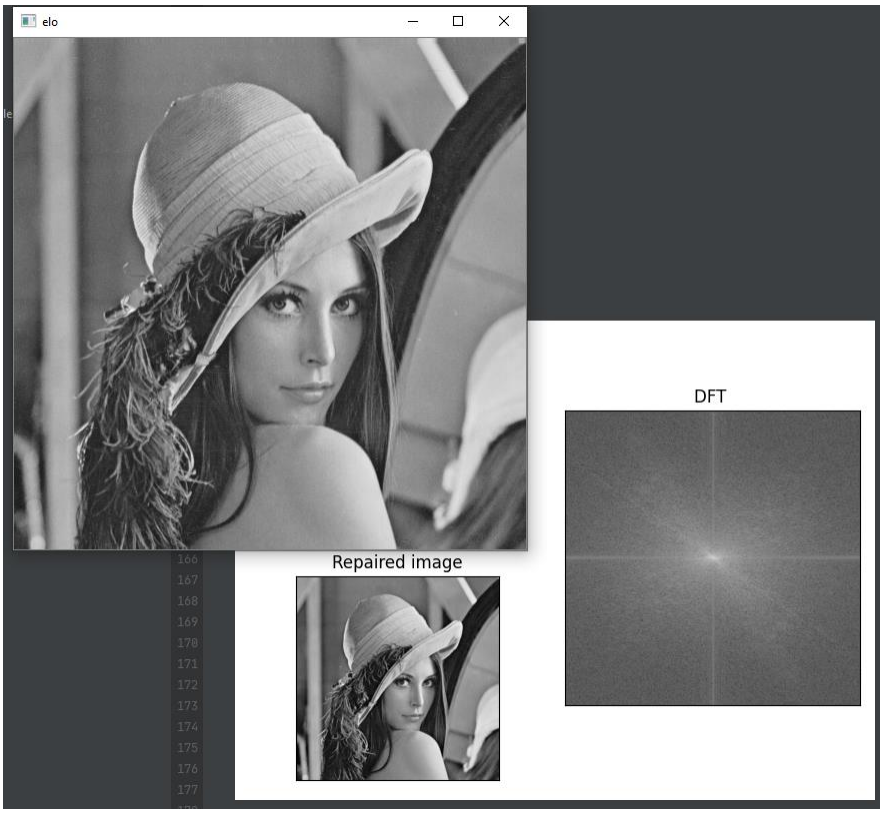
However, a histogram of thresholded image differs from them. This occurs, because in thresholding a value of a pixel can take only one of two values, 0 or 255, which is visible on the graph in two bars.

- **Task 4**

In order to realize task 4 I did some research with the OpenCV documentation and with use of it I have implemented DFT() function which realizes discrete Fourier transform and inverted DFT of the given image. The body of the function is mostly based on the OpenCV documentation however I have tired to improve it a bit.

```
def DFT(img):
    """Tranforms the given image into the frequency domain with use of DFT.

        Parameters
        ----------
        img : OpenCV type
            The image to be processed

        """
    dft = cv.dft(np.float32(img), flags=cv.DFT_COMPLEX_OUTPUT)
    dft_shift = np.fft.fftshift(dft)
    magnitude_spectrum = 20 * np.log(cv.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))

    plot.subplot(122), plot.imshow(magnitude_spectrum, cmap='gray')
    plot.title('DFT '), plot.xticks([]), plot.yticks([])

    row = img.shape[0]
    col = img.shape[1]
    rrow = row // 2
    ccol = col // 2

    mask = np.ones((row, col, 2), np.uint8)
    mask[rrow:rrow, ccol:ccol] = 0

    fshift = dft_shift * mask
    f_ishift = np.fft.ifftshift(fshift)

    res = cv.idft(f_ishift)
    res = cv.magnitude(res[:, :, 0], res[:, :, 1])

    plot.subplot(223), plot.imshow(res, cmap='gray')
    plot.title('Repaired image'), plot.xticks([]), plot.yticks([])

    plot.show()
```

Note: In order to use OpenCV function **dft()** we have to convert image into float32 type.



Personally I would say the repaired image is a bit sharper then the original one but it might be just an illusion.

Summary

During this labs we learned many functions very useful in image processing and we learned more about histograms, we discovered quantization, thresholding and Discrete Fourier Transform.