CS-484

Fall 2024

Homework Assignment 1

Name: Ahmet Bera Özbolat

ID: 21902777

1. Morphological Operations

In this task, the goal is to isolate the circles in the image by making the background completely black so that only the circles are white. To achieve this, I have applied erosion first and dilation later known as opening.

Erosion: Erosion removes pixels from the object boundaries and small noises and irrelevant details like small white parts by shrinking the white regions.

Dilation: The dilation method is applied to restore the image and conserve the black ground. Dilation enlarges the white areas and compensates for shrinkage caused by dilation

Kernel Size**:** In order to erode and dilate the image perfectly choosing Kernel size is important otherwise erosion can completely erode white circles or erosion may not erode the white regions enough to get rid of the noise or corrupted parts.

I have used the same Kernel size for both operations to compensate for shrinking and enlarging side effects. I have used 3x3, 7x7, 100x100 respectively results can be found in the figures below.

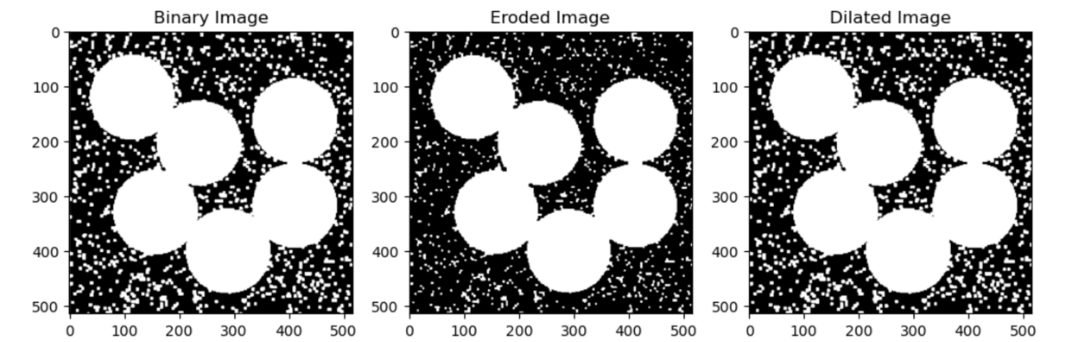

Figure 1:Kernel size=3x3

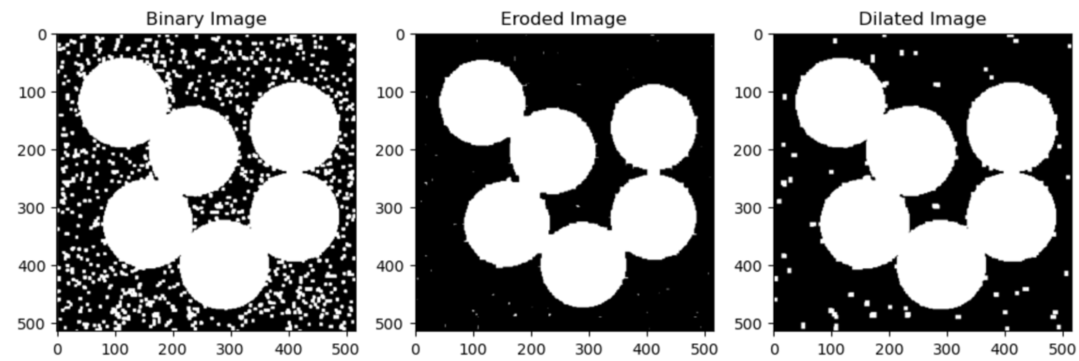

Figure 2: Kernel size=7x7

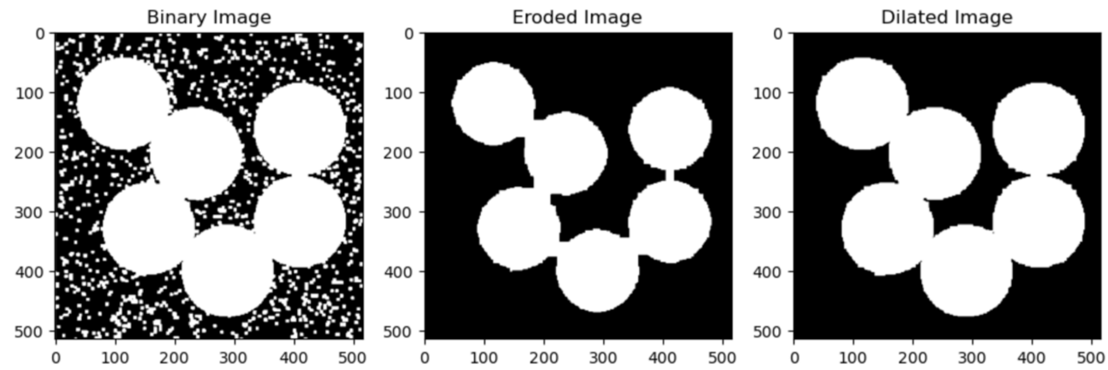

Figure 3: Kernel size=15x15

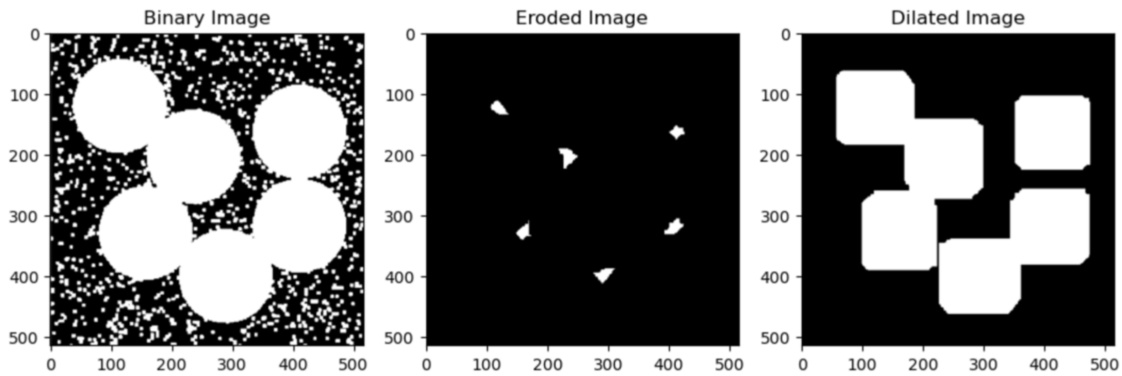

Figure 4: Kernel size=100x100

As we can see in Figure 1 and Figure 2, too small kernel sizes cannot clear off white areas properly and Figure 4 shows that too big kernel sizes may cause over shrinkage and enlargments. Appropriate Kernel size such as in Figure 3 accomplishes the task completely.

2. Histogram-Based Image Enhancement

Part 1:

In this part, the goal was to implement a custom histogram function and compare the results with built-in function results. I first take the image as a 2D matrix then I flatten the matrix into a 1D array in this way I have computed the frequency of each intensity level easily. I have also computed the histogram of the image by using the “cv2” library, as can be seen in the figures below results were consistent with the custom function results confirming that implementation was successful.

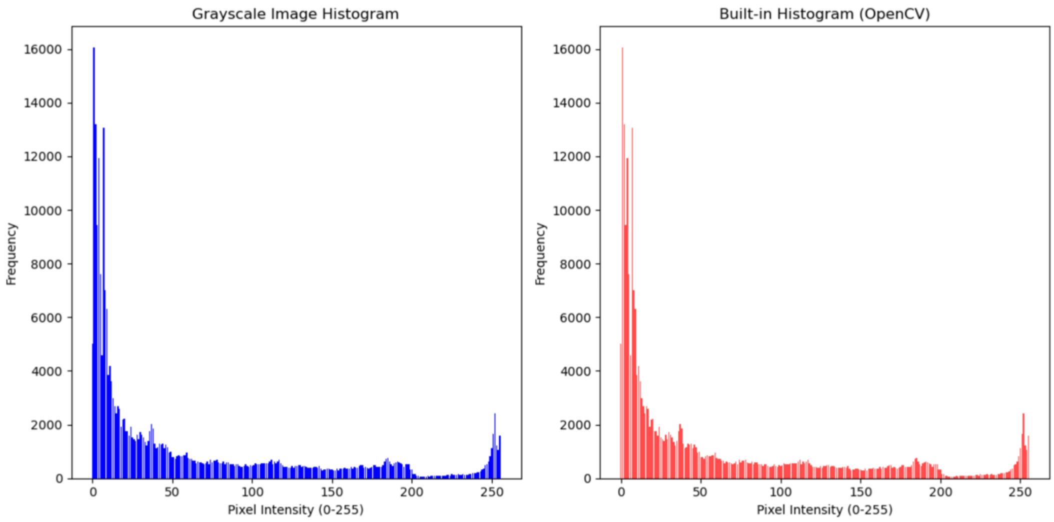

Figure 5:Histogram results of first image

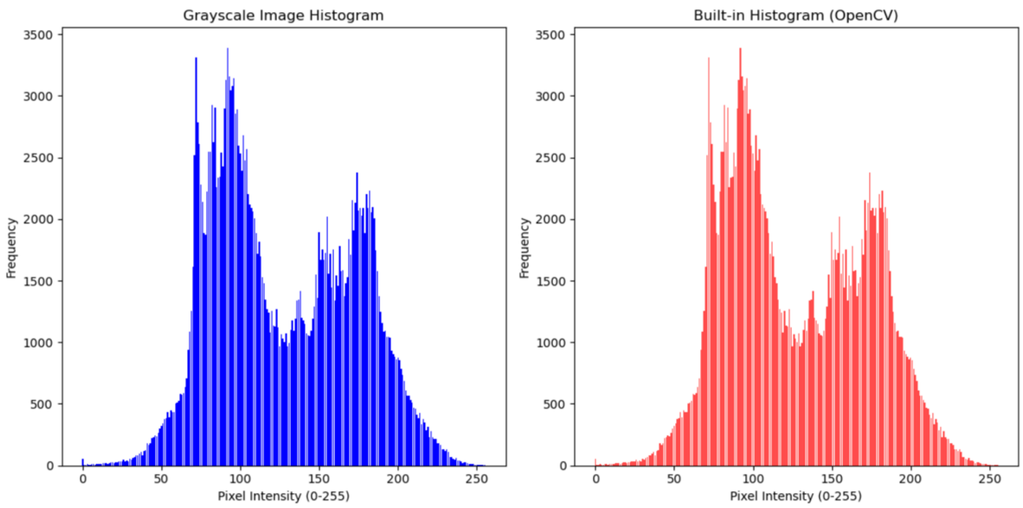

Figure 6:Histogram results of second image

Part 2:

In this part goal is to implement contrastive stretching on an image, I have first extracted the minimum and maximum pixel intensity values from the image and applied the provided formula for transforming the image into new intensity ranges. Afterward, I plotted the results with different [c,d] ranges.

[c,d]:

0-128 => Resulted in a darkest image since the range of pixel intensity levels was mapped to lower, therefore, the result were high contrast version of the original image

128-255=> Resulted in the whitest image since intensity ranges were mapped to higher levels therefore, the results were a low-contrast version of the original image.

0-255=> Resulted in a grayish image because of the full intensity range therefore balanced contrasted version of the image.

Figure 7: 0-128 range

Figure 8: 128-255 range

Figure 9:0-255 range

3. Otsu Thresholding

In this part aim was to apply Otsu Thresholding to two different images, The Otsu Method separates image foreground and background by finding the optimal threshold that minimizes the intra-class variance or maximizes the between-class variance algorithm iterates all possible threshold values and checks which threshold fits the criteria best.

I have first used my previous histogram function to find pixel intensities of the image so that the distribution of intensities can be analyzed. Then applying the algorithm in Figure 10, I found the best threshold value by iterating that maximizes between-class variance.

variance, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Weights ω_0 and ω_1 are the probabilities of the two classes separated by a threshold t ,and σ_0^2 and σ_1^2 are variances of these two classes.

The class probability $\omega_{\{0,1\}}(t)$ is computed from the L bins of the histogram:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i)$$

For 2 classes, minimizing the intra-class variance is equivalent to maximizing inter-class variance:[2]

$$\begin{aligned}\sigma_b^2(t) &= \sigma^2 - \sigma_w^2(t) = \omega_0(t)(\mu_0 - \mu_T)^2 + \omega_1(t)(\mu_1 - \mu_T)^2 \\ &= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2\end{aligned}$$

which is expressed in terms of class probabilities ω and class means μ, where the class means $\mu_0(t)$, μ

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)}$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)}$$

$$\mu_T = \sum_{i=0}^{L-1} ip(i)$$

The following relations can be easily verified:

$$\begin{aligned}\omega_0\mu_0 + \omega_1\mu_1 &= \mu_T \\ \omega_0 + \omega_1 &= 1\end{aligned}$$

Figure 10: Algorithm from wikipedia(alıntı)

Afterward, I applied the algorithm to both images and plotted the results which can be in the figures below.

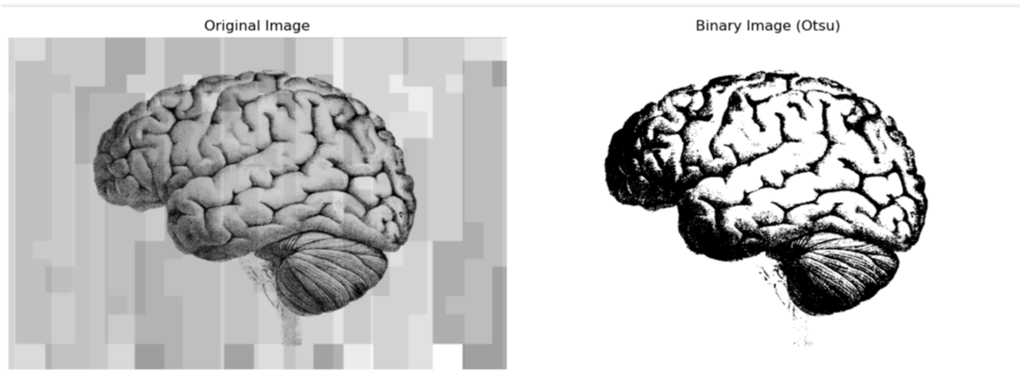

Figure 11:First image Otsu result threshold=120

Figure 12:Second image Otsu result threshold=152

For the first image separation between foreground and background was successful with small errors However, in the second image some areas in the foreground separated as background this happened due to the distribution of pixel intensities because the otsu method assumes bimodal histogram when there are no two peaks in the histogram distribution, model cannot perform well.

4. 2-D Convolution in Spatial and Frequency Domain

Part 1:

In this part, the aim was to implement a convolution function in the spatial domain and use it for edge detection by using the Sobel and Prewitt operators. I have first implemented a convolution function since it is similar to the work in the first task I have used some parts of my code for the first part. Afterward, I applied convolution to the image using their specific kernels from the x and y directions and used the results to find edges. The difference between Sobel and Prewitt operators is in their kernels in practice however, their focus is different.

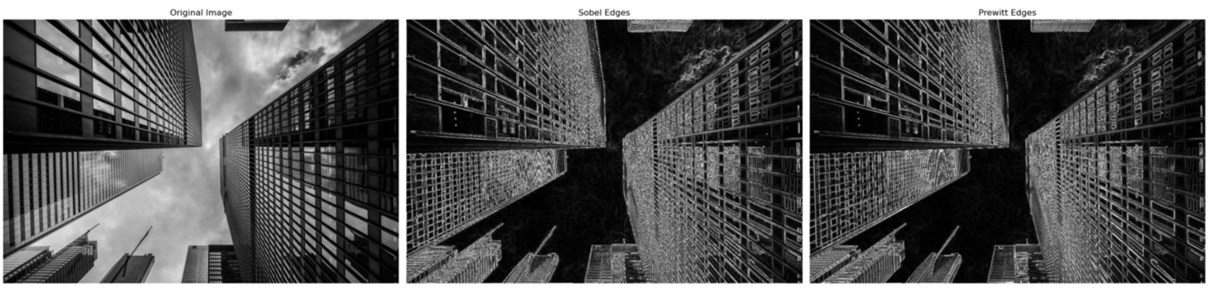

Figure 13:Sobel and Prewitt Operator Results

As Figure 13 shows, the Sobel operator has better noise resistance, more sensitive to the diagonal edges than vertical and horizontal edges on the other hand, the Prewitt operator is more sensitive to the horizontal edges and vertical edges focus on the edges more but more sensitive to the noise.

Part 2:

In this section, the aim was to implement frequency domain filtering on a grayscale image using a low-pass Gaussian filter. First of all, I created my Gaussian low-pass filter and then using the hint from lecture notes I followed the steps. I started with centering the image and then I took the DFT of the centered image then I multiplied my custom Gaussian filter then took the real part of IDFT of the multiplication of the recentered image.

Afterward, I applied filtering to the image and realized that its the opposite of spatial domain filtering as I increased the variance of the Gaussian filter blurring started to decrease opposite of what I was expecting after some research this happened because when we apply low-pass filtering in the frequency, Gaussian filter with low variance filter becomes narrow therefore only low-frequency levels are passing higher frequencies are filtered therefore image appears to be smoother and blurry on the other hand at higher variance levels allows to pass high frequency which corresponds to edges and details.

The result of the difference in variance levels can be seen in the figures below. As the figures show after some point increase in variance does not effect the result since most of the frequencies are not filtered.

Figure 14:LPF sigma=5

Figure 15: LPF sigma=10

```
deneme.display_results()
```

Figure 16:LPF sigma=100

Figure 17:LPF sigma=1000

References

1-7.3. Padding and Stride — Dive into Deep Learning 1.0.3 documentation. (n.d.). https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html

2-Bin, L., Samiei Yeganeh, M., & School of Information Technology, Jawaharlal Nehru Technological University, China, Iran. (2012). Comparison for Image Edge Detection Algorithms. In *IOSR Journal of Computer Engineering* (Vol. 2, Issue 6, pp. 01–04). https://www.iosrjournals.org/iosr-jce/papers/vol2-issue6/A0260104.pdf

3-Image Filtering in the Frequency Domain. (n.d.). https://paulbourke.net/miscellaneous/imagefilter/

4-OpenCV: Morphological Transformations. (n.d.). https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

5-<i>Types of Morphological Operations - MATLAB & Simulink. (n.d.). <span class=&qout;url&qout;>https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html

6-Wikipedia contributors. (2024a, April 17). *Otsu's method*. Wikipedia. https://en.wikipedia.org/wiki/Otsu%27s_method

7-Wikipedia contributors. (2024b, August 30). *Gaussian filter*. Wikipedia. https://en.wikipedia.org/wiki/Gaussian_filter