CS484/555
Fall 2024
Homework Assignment 1

Due: 28 October 23.59

Morphological Operations [20 pts]

In this question, you will write your own implementations of the two fundamental morphological operations, namely ***dilation*** and ***erosion***. You will write a separate function for each operation. These functions will take as input a binary image (as a matrix) and a structuring element (also as a matrix), and produce a binary image (another matrix) corresponding to the result of the operation.

- *dilated image = dilation(src img, struc. element)*
- *eroded image = erosion(src img, struc. element)*

You should generate the structuring element as a binary image with an arbitrary shape. Given the structuring element, your code should implement the dilation and erosion operations using the definitions given in the course. Note that the structuring element should be created (as a matrix) outside and given as input to the dilation/erosion functions so that your code can work with any kind of structuring element. After your implementation, for the image in Fig 1 (this image is separately provided to you as a part of this homework), you are expected to apply a sequence of morphological operations so that only the circles at the middle are white and the background is completely black. In the end, you should have white circles with approximately the same size as the initial image on a black background. Explain the order of operations that you have executed and the reasoning behind it in your report.

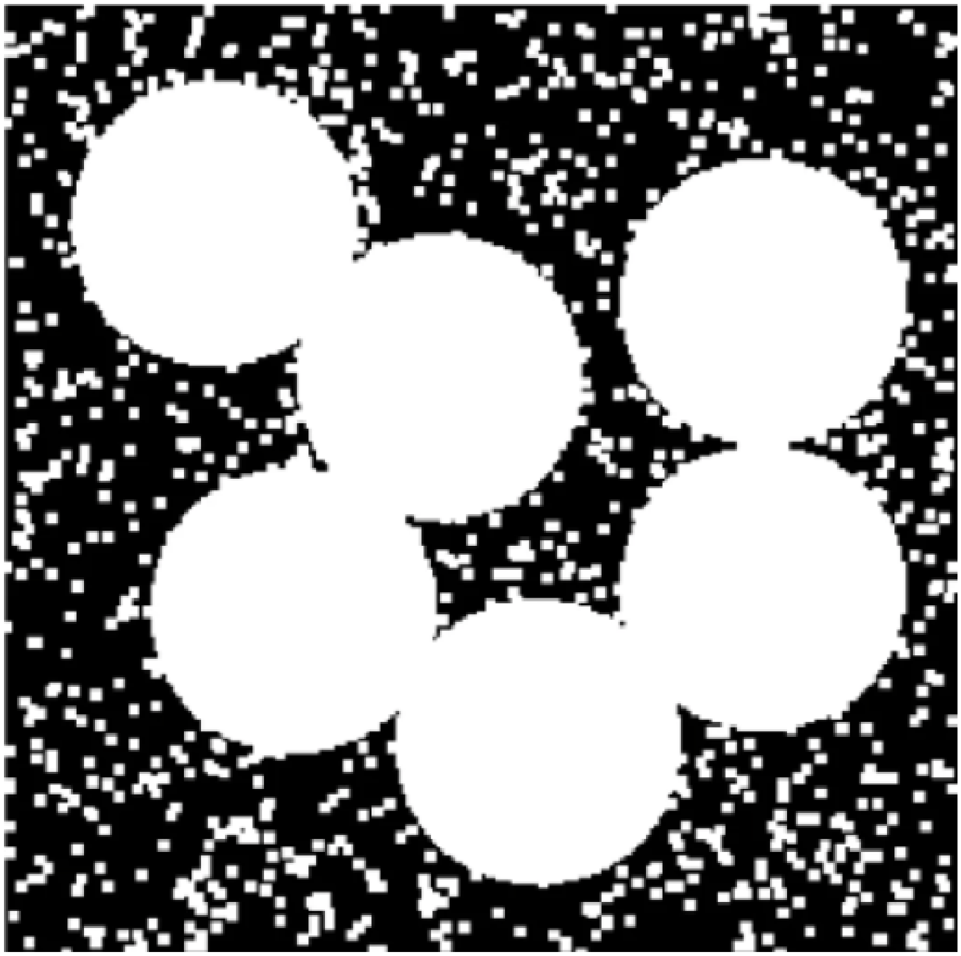

Figure 1: Circles with noisy background.

Histogram-Based Image Enhancement [20 pts]

In this part of the homework, you need to implement 2 different functions with the following specifications.

Part 1 [10 pts]

Implement a function to generate a histogram of a grayscale image. The prototype of the function should be the following:

- *histogram(source_image)*

Your implementation should be generic for any given grayscale image. That is, it should be able to work with images of any size. Note that, you can use additional built-in functions or libraries to read an image as a matrix and for plotting histograms. However, your function should only take the image matrix as input. No image-processing libraries are to be used for the implementation of the histogram function. Please test your results on the images provided in Fig. 2 (both of these images are also provided to you separately as a part of this homework) and include it in your report. (Hint: compare your results with MATLAB's built-in histogram function.)

Part 2 [10 pts]

Implement a function that implements contrastive stretching on an image:

- *contrast_stretching(source_image)*

Given a grayscale image, implement a contrast stretching function. Contrast stretching is a simple image enhancement technique used to improve the contrast in an image by stretching the range of intensity values. The transformation function can be defined as:

$$f(x) = \frac{(x-a)}{(b-a)} \times (d-c) + c$$

Where:

- x is the pixel intensity in the original image.
- a and b are the input image's minimum and maximum pixel intensity values, respectively.
- c and d are the desired minimum and maximum pixel intensity values in the output image, respectively.

Your function should do the following:

1. get a grayscale image.
2. Identify the minimum and maximum intensity values a and b in the input image.
3. Stretch the pixel intensities using the formula above to map them to a new range $[c, d]$

Write a test program that loads a grayscale image in Figure 3 (this image is also separately provided to you as a part of this homework) and applies the contrastive stretching idea for [c,d] values of $(0, 255)$, $(128, 255)$ and $(0, 128)$. Explain how different c and d values have effected the final output in your report.

Figure 2: Images for histogram generation.

Figure 3: Image for contrast stretching.

Otsu Thresholding [20 pts]

Implement automated thresholding by Otsu's Method. As in the first question, you must use your own implementation and avoid using any image processing libraries (except for reading and plotting images). Your function declaration should take a grayscale image (a 2D matrix) as input and produce a binary image (also a 2D matrix) of the same dimensions as the input image. Use the format below:

- binary image = otsu threshold(source image)

Separate the background from the foreground using your implementation of Otsu's algorithm. Show your results on two different images shown in Figure 4 (both of those images are also provided as a part of this homework). Discuss your results. Are they always perfect? Why so, why not?

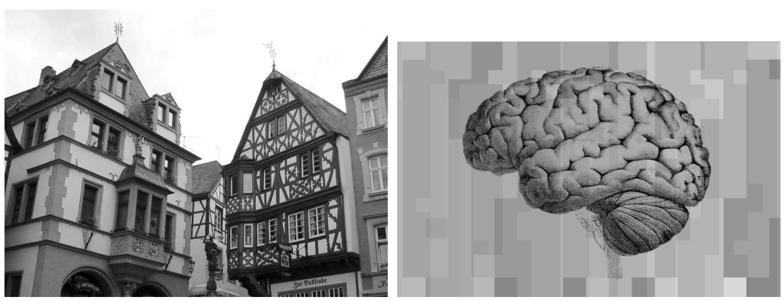

Figure 4: binary image = otsu threshold(source image)

2-D Convolution in Spatial and Frequency Domain [30 pts]

Part 1 [10 pts]

In this question, you are going to implement a two-dimensional convolution operation in the spatial domain. The function you write for convolution should take as input an image as its first argument and the filter as its second argument and output the result of convolving the image with the given filter in the spatial domain. Your function should be generic to accept any given filter or image. However, we assume that the images are represented as 2D matrices, i.e. multi-channel images (e.g. colour images) are not allowed. Likewise, filters are also 2D matrices. While implementing your function, take care of boundaries. Use your convolution function for edge detection using the Sobel and Prewitt operators for the image in Fig 5. Explain how Sobel and Prewitt operators have affected the output of the image. Which component of the image do they focus on more?

Part 2 [20 pts]

In this section, you are tasked with performing frequency domain filtering on a grayscale image using a low-pass Gaussian filter. You are required to implement the filtering process by applying the Convolution Theorem: transform the image into the frequency domain, apply the low-pass Gaussian filter, and subsequently transform the result back into the spatial domain. No pre-defined filter will be provided, so you must design and implement the low-pass Gaussian filter independently.

Test your filtering algorithm on the image provided in 6 with a Gaussian filter of a sufficiently large radius so that the image is blurred. Display both the original image and the filtered image and provide a detailed explanation of the methodology you followed, including the steps taken during the transformation, filtering, and reconstruction phases in your report. (Hint: You can use the information provided on your lecture slides in the second part of chapter 3)

Figure 5: Image for the Sobel and Prewitt filters.

Figure 6: Image for Low Pass Gaussian filter

Submission

Your submission should include

- A report (pdf file) that includes the results from all the questions, your discussions, generated plots and etc. Remember that slightly imperfect results are quite likely due to the limitations of the methods. As long as you are able to provide an analytical explanation for such results, and as long as these results are similar in quality to results from MATLAB/PYTHON libraries, you are OK. In fact, you are encouraged to compare your implementation to those of MATLAB/PYTHON libraries, but please DO NOT INCLUDE these comparisons in your report! In parts of the assignment where you are allowed to use external libraries, specify the library and the method that you are using.
- A script that runs the particular sequence of operations and reproduces the result presented in your report.
- You will upload a SINGLE zip file on Moodle

For any questions, please contact yigit.ekin@bilkent.edu.tr via email.