CS 484/555
Spring 2024
Homework Assignment #1

Due: 27th March 23:59

Question 1 [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.

$dilated\_image = \ dilation(src\_img, struct\_et)$

$eroded\_image = \ erosion(src\_img, struct\_et)$

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 codes so that your code can work with any kind of structuring element.

After your implementation, for the image given below, you are expected to apply a sequence of morphological operations to come up with a cleaner version of the image given in Fig. 1 where the noise is removed and the characters are readable. (this image is also provided as a part of this homework)

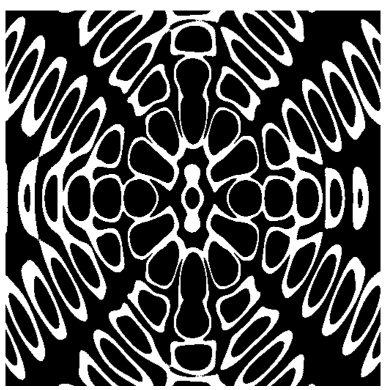

Figure 1: diffraction pattern

Question 2 [10 pts] Implement a function to generate a histogram of a grayscale image. Your method should take an arbitrary grayscale image (a matrix) as an input and plot the histogram of pixel intensities. Your implementation should be generic for any given grayscale image, meaning 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 an input. No image-processing libraries are to be used for the implementation of the histogram function. Test your results on the images provided below. Compare your results with MATLAB's built-in histogram function.

(a) First grayscale image

(b) Second grayscale image

Figure 2: Images for Histogram Generation

Question 3 [20 pts] Implement the histogram equalization technique as discussed during the lectures to improve the contrast in an image. The method should read the histogram of a given greyscale image, equalize it, and output the enhanced image. This procedure aims to achieve a (almost) uniform distribution of intensities, improving the visual appearance of the image. Apply your implementation of the histogram equalisation on the two images in Fig. 2. Does your implementation of histogram equalisation yield an exactly *uniform* distribution? why/why not?

Question 4 [10 pts] Implement automated thresholding by Otsu's Method. 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 an input and produce a binary image (also a 2D matrix) of the same dimensions as in 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 3 (both of those images are also provided as a part of this homework). Discuss your results, are they always perfect? Why so, why not?

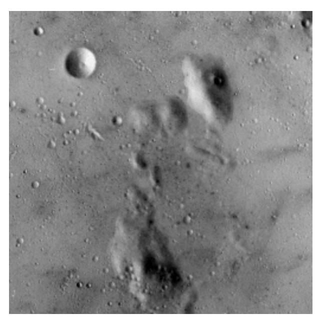

Figure 3: Images for Otsu's threshold

Question 5 [20 pts] In this question, you will use your previously implemented functions. You are expected to apply a sequence of thresholding and morphological operations to label and count distinct objects in the given image Figure 4 individually.

Objective

The goal is to identify and count the distinct objects in each provided grayscale image through a series of image processing steps. The process involves thresholding to separate objects from the background, morphological operations to refine the object shapes, and connected components labeling to label distinct objects.

Procedure

1. **Thresholding:** Find a threshold that produces a binary image, effectively separating objects from the background. Specify the method used for determining the threshold (e.g., trial and error or Otsu's method) and justify your choice.

2. **Morphological Operations:** Use the implemented morphological operators (dilation, erosion, and their combinations) to separate objects connected together or to fill holes within the objects. Consider the use of inverse operations on the binary image, especially if the objects are lighter than the background.

3. **Labelling:** With a refined binary image, employ connected components analysis to produce a labelled image, assigning a unique label (an integer ID) to each distinct region.

Restrictions: You MUST use your own implementations for dilation, erosion, and thresholding. However, you CAN use external libraries (such as OpenCV) for arithmetic and logical operations, connected components labelling, and image I/O functions. You are NOT allowed to use any advanced functions beyond the specified operations.

Analytical Justification

Provide an analytical justification for the sequence of operations performed. Explain what each step achieves in the context of object labelling and counting. It is essential to not only provide a working sequence but also to explain the rationale behind each step. You are encouraged to experiment with different sequences of operations to achieve the desired result but report only one such sequence.

Figure 4: Corn seeds

Question 6 [20 pts] In this question, you are going to implement a two-dimensional convolution operation. 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 the boundaries. Use your convolution operator for edge detection using the Sobel and Prewitt operator for the image in Fig. 5.

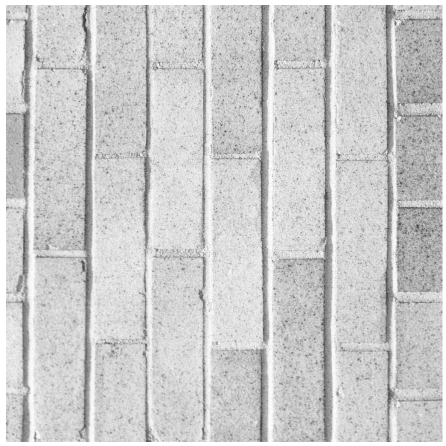

Figure 5: Image for Sobel and Prewitt filter

Submission:

- A report (pdf file) that includes the results from all 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 *sepehr@bilkent.edu.tr* via email.