Lab 4 - Morphological operations

- Morphological operations on binary images and on gray-level images.
- Dilation and Erosion.
- Opening and Closing.
- Region segmentation and Flood-Filling

4.1 Binary images — Dilation

When applied to binary images, the morphological dilation operation expands the boundaries of foreground regions.

Given the gray-level image **wdg2.bmp**, create a new program (**Aula_04_exe_01.py**) carrying out the following sequence of operations:

- Conversion to a binary image, with threshold 120.
- Inversion of the resulting image (i.e., obtaining the negative image).
- Dilation of the negative image using a circular structuring element, with a diameter of 11 *pixels*.

What happens if you repeatedly apply the dilation operation using the same structuring element?

Now, use a square structuring element, of size 11×11. Repeatedly apply the dilation operation. What differences do you notice?

4.2 Edge detection with morphological operations

The morphological dilation can be used to obtain image edges.

Given the gray-level image **wdg2.bmp**, carry out the following sequence of operations:

- Conversion to a binary image, with threshold 120.
- Inversion of the resulting image (i.e., obtaining the negative image: Image A).
- Dilation of the negative image using a square structuring element of size 3×3.
- Subtration of Image A form the resulting dilated image. subtrair a imagem a a imagem square

Carry out the same sequence of operations using a larger structuring element. What differences do you notice?

4.3 Binary images — Erosion

When applied to binary images, the morphological erosion operation essentially shrinks the boundaries of foreground regions.

Given the gray-level image **wdg2.bmp**, carry out the following sequence of operations:

- Conversion to a binary image, with threshold 120.
- Inversion of the resulting image (i.e., obtaining the negative image).
- Erosion of the negative image using a circular structuring element, with a diameter of 11 *pixels*.

What happens if you repeatedly apply the erosion operation using the same structuring element?

Now, use a square structuring element, of size 11×11. Repeatedly apply the erosion operation. What differences do you notice?

The morphological erosion has directional effects, when using non-symmetrical structuring elements.

Try using:

- A structuring element of size 11×1.
- A square structuring element of size 3×3; but with its origin (*"hotspot"*) in the center pixel of the first row.

What happens?

4.4 Segmentation with morphological operations

A morphological erosion might be a first step before segmenting contiguous image regions.

Given the gray-level image **mon1.bmp**, carry out the following sequence of operations:

- Conversion to a binary image, with threshold 90.
- Inversion of the resulting image (i.e., obtaining the negative image).
- Repeated erosion (twice) of the resulting image using a circular structuring element, with a diameter of 11 *pixels*.

What happens if you use a square structuring element of size 9×9?

4.5 Opening

The morphological opening operation corresponds to applying an **erosion** operation followed by a **dilation** operation, using the same structuring element.

Given the binary image **art3.bmp**, we want to count the circular regions. Carry out a morphological opening using a circular structuring element, with a diameter of 11 *pixels*.

Given the binary image **art2.bmp**, we want to separately segment the vertical and the horizontal line segments. Carry out a morphological opening using a rectangular structuring element of size **3×9**, and using a rectangular structuring element of size **9×3**. What happens?

4.6 Closing

The morphological closing operation corresponds to applying a **dilation** operation followed by an **erosion** operation, using the same structuring element.

Given the binary image **art4.bmp**, we want to remove the circular regions of smaller size. Carry out a morphological closing using a circular structuring element, with a diameter of 22 *pixels*.

Use structuring elements of smaller and larger diameter. Analyze the resulting images.

4.7 Region Segmentation using Flood-Filling

Create a new example (**Aula_03_exe_07.py**) that allows segmenting regions of a given image.

Starting from a **seed pixel**, the **floodFill** function segments a region by spreading the seed value to neighboring pixels with (approximately) the same intensity value.

Use the function

```
retval, rect = cv2.floodFill(image, mask, seedPoint, newVal[, loDiff[, upDiff[, flags]]])
```

Segment the **lena.jpg** image, using as a seed the pixel **(430, 30)** and allowing intensity variations of ±5 regarding the intensity value of the seed pixel.

Optional

Allow the user to interactively select the seed pixel for region segmentation.

Test the interactive region segmentation using the wdg2_ bmp, tools 2_png and lena_ jpg images.