Final Project Report

# Introduction:

Morphology is a broad set of image processing operations that process the input images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. The structuring element can of various types like square, cross etc. In a morphological operation, the value of each pixel in the output image depends on a comparison of the corresponding pixel in the input image with its neighbors. Morphological operations are best suited for binary images. However, they can also be performed on greyscale images.

Dilation, erosion, open, close, open-close, close-open, skeletonization**(have to change the list late)** are some of the morphological operations that can be performed on the images.

Our aim is to perform the above listed morphological operations on the images by altering the size of structuring element and analyze its effect on binary and grayscale images.

## How the operations are performed?

For the above mentioned operations Erosion, Dilation, Open, Close, Open-Close, Close-Open; allow the user to perform required operation on the specified image. For that purpose we have created an interface so that the user can upload any image of his/her choice. Then select the input image type which is binary or gray scale. And the user also has to select the type of the output image which can either be in binary or grayscale.

Then the user can select the operation he/she wishes to perform on the image from the dropdown provided in the interface. It can be anything from the list containing Erosion, Dilation, Open, Close, Open – Close, Close – Open. Then, clicking on the submit button performs the given operation and outputs the image in the user preferred type.

The following are the operations performed

## Dilation:

Dilation adds pixels to the boundaries of objects in an image. Number of pixels added to the objects in an image depends on the size and shape of the *structuring element* used to process the image. An equation to represent dilation is

Given a window **B** and a binary image **I**: **J1 = DILATE(I,B)** if J1(I,j) = **OR**{**B**.**I**(I,j)} = **OR**{**I**(i-m,j-n); (m,n) ε **B**}

### Binary Input Image and Binary Output Image:

When the dilation operation is performed on the user input binary image, the output image in binary format looks as shown below.

### Grayscale Input Image and Binary Output Image:

When the dilation operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the dilation operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below.

## Erosion:

Erosion removes pixels to the boundaries of objects in an image. Number of pixels removed from the objects in an image depends on the size and shape of the *structuring element* used to process the image. An equation to represent erosion is

Given a window **B** and a binary image **I**: **J2= ERODE(I,B)** if J1(I,j) = **AND**{**B**.**I**(I,j)} = **AND**{**I**(I-m,j-n); (m ,n) ε **B**}

### Binary Input Image and Binary Output Image:

When the erosion operation is performed on the user input binary image, the output image in binary format looks as shown below

### Grayscale Input Image and Binary Output Image:

When the erosion operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the erosion operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below

## Open:

Open operation is erosion followed by dilation using the same structuring element for both operations. Opening is the dual of closing *i.e* opening the foreground pixels with particular structuring element is equivalent to closing the background pixels with the same element.

Given an image I and window B, opening operation can be represented as

OPEN(I, B) = **DILATE** [**ERODE**(I, B), B]

### Binary Input Image and Binary Output Image:

When the open operation is performed on the user input binary image, the output image in binary format looks as shown below

### Grayscale Input Image and Binary Output Image:

When the open operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the open operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below

## Close:

Closing is opening performed in reverse. It is defined as dilation followed by erosion using the same structuring element for both operations. Closing is the dual of opening *i.e* closing the foreground pixels with particular structuring element is equivalent to opening the background pixels with the same element.

Given an image I and window B, opening operation can be represented as

CLOSE(I, B) = **ERODE** [**DILATE** (I, B), B]

### Binary Input Image and Binary Output Image:

When the close operation is performed on the user input binary image, the output image in binary format looks as shown below

### Grayscale Input Image and Binary Output Image:

When the close operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the close operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below

## Open-close:

Open-close is very effective smoother that can be obtained by sequencing the OPEN and CLOSE operators. OPEN-CLOS tends to link neighboring objects together.

For an image I and structuring element B, the equation can be defined as:

OPEN-CLOSE(**I**, **B**) = **OPEN** [**CLOSE** (**I**, **B**), **B**]

### Binary Input Image and Binary Output Image:

When the open-close operation is performed on the user input binary image, the output image in binary format looks as shown below

### Grayscale Input Image and Binary Output Image:

When the open-close operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the open-close operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below

## Close-open:

Close-Open is very effective smoother that can be obtained by sequencing the OPEN and CLOSE operators. CLOS-OPEN tends to link neighboring holes together.

For an image I and structuring element B, the equation can be defined as:

CLOSE-OPEN(**I**, **B**) = **CLOSE** [**OPEN** (**I**, **B**), **B**]

### Binary Input Image and Binary Output Image:

When the close-open operation is performed on the user input binary image, the output image in binary format looks as shown below

### Grayscale Input Image and Binary Output Image:

When the close-open operation is performed on the user input grayscale image, the output image in binary format looks as shown below.

### Grayscale Input Image and Grayscale Output Image:

When the close-open operation is performed on the user input grayscale image, the output image in grayscale format looks as shown below

## Skeletonization:

Skeletonization is a process for reducing foreground regions in a [binary image](http://homepages.inf.ed.ac.uk/rbf/HIPR2/binimage.htm) to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels.

**(Should include a formula here)**

## Boundary extraction:

Boundaries are linked edges that characterize the shape of an object. They are useful in computation of geometry features such as size or orientation.

Boundary extraction can be simply obtained by erosion operation. Initially an erosion operation has to be performed on the image and the obtained result has to be subtracted from the original image to extract the boundary of the original image.

## Majority:

Majority operation possesses qualitative attributes of both dilation and erosion, but does not generally change the size of objects or background. The median removes the small objects and small holes, but did not change the boundary (size) of the image on the whole.

Given a window B and a binary image I, the equation can be defined as:

J3 = MEDIAN(**I**, **B**) if J3 (i, j) = MAJ{**B** • I(i, j)} = MAJ{**I**(i-m, j-n); (m, n) ∈ **B**}

# Analysis:

1. Structing shape
2. Structuring size
3. Iterations