# HOMEWORK 1 U2
# MORPHOLOGICAL OPERATIONS 1
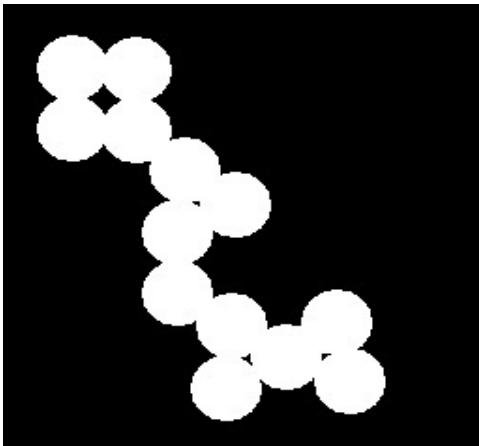
## Héctor Mauricio Mendoza Xicoténcatl

**Abstract**—In this report, we documented the first of three homework assignments related to morphological transformations. Particularly, in this activity we used two important OpenCV commands: Dilation and Erode.

**Index Terms**—OpenCV, Erode, Dilate, Bitwise-or, Morphology.

✦

## 1 PROBLEM TO SOLVE

From the following image:



- Find the skeleton of the image.
- Fill the image until you obtain the circles with no black spaces between them.

## 2 CODING PROCESS

In this section, we coded a single file that involves the two exercises requested. We will also explain the most important code lines that help us to solve the problems.

### 2.1 SKELETON & DILATION

Similar to previous exercises, first we imported the libraries necessary for the program with the commands:

```
import numpy as np
import cv2
```

We read the circle image:

```
img_org = cv2.imread('circles.jpg')
img = cv2.imread('circles.jpg', 0)
```

We create an empty skeleton of image:

```
size = np.size(img)
skel = np.zeros(img.shape, np.uint8)
kernel = np.ones((5,5), np.uint8)
```

In the last line, we defined our kernel matrices that later will be useful to eliminate the black spaces in our circles figure.

```
elem = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
dil_img = cv2.dilate(img, kernel, iterations=2)
```

The first line above gets a Cross Shaped Kernel. The last one is in charge to eliminate the black spaces between circles.

```
    while True:
—-    open = cv2.morphologyEx(img, cv2.MORPH_OPEN, elem)
—- temp = cv2.subtract(img, open)
—- eroded = cv2.erode(img, elem)
```

**—- kel = cv2.bitwise_or(kel,temp)**
**—- img = eroded.copy()**
**—- if cv2.countNonZero(img)==0:**
**——- break**

The cycle while above erodes the original image and refine the skeleton.
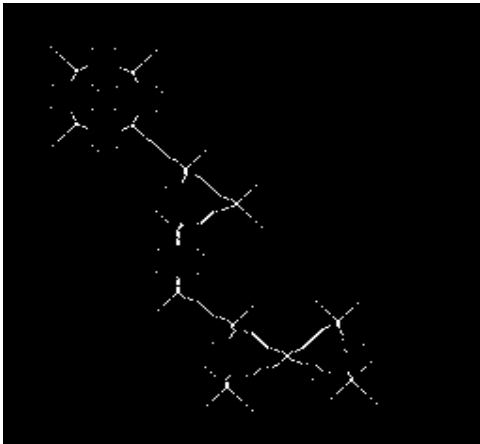Finally, we print the results:

**cv2.imshow("Original", img_org)**
**cv2.imshow("Skeleton", kel)**
**cv2.imshow('Dilation', dil_img)**
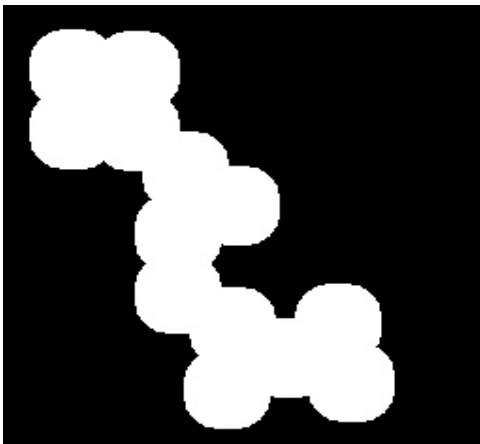
# 3 RESULTS:



**Skeleton Results**



**Dilation Results**

this activity is another tool we have learned to use in OpenCV. I consider this activity will be some sort of useful in future morphological operation activities. The implementation was relatively easy, given the fact, there is existing documentation of the topic and even more, of the function to be used in OpenCV.

**Héctor Mendoza** Héctor Mauricio Mendoza Xicoténcatl is a Computational Robotics Junior student, currently stu-diying Universidad Politécnica de Yucatán.

# 4 CONCLUSIONS

In this section, we coded a single file that involves the two exercises requested. We will also explain the most important code lines that help us to solve the problems. We can say that