IPPR Lab 7

Name : Arya Shah

Roll No. E071

Class : BTech CSBS

Aim: To Apply Erosion,Dilation and Opening Operations On The Given Test Image

```
# Importing Necessary Libraries
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from skimage.color import rgb2gray, rgba2rgb
import cv2
```

To Apply Erosion, Dilation & Opening operation of given image.

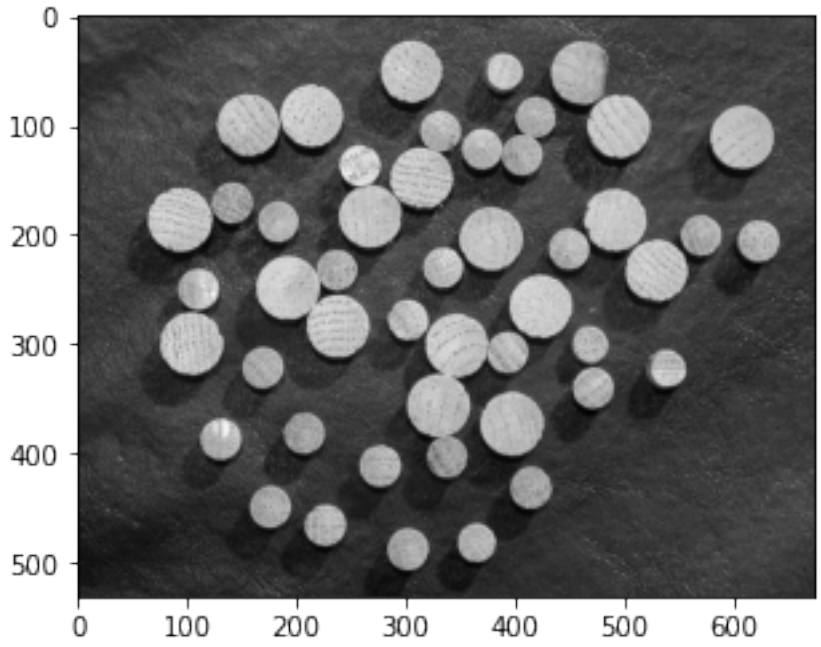
```
image = io.imread('wood_disk.tif') #Read pool.png File
```

```
sh = image.shape
sh
```

```
(531, 675)
```

```
plt.imshow(image, cmap='gray')
```

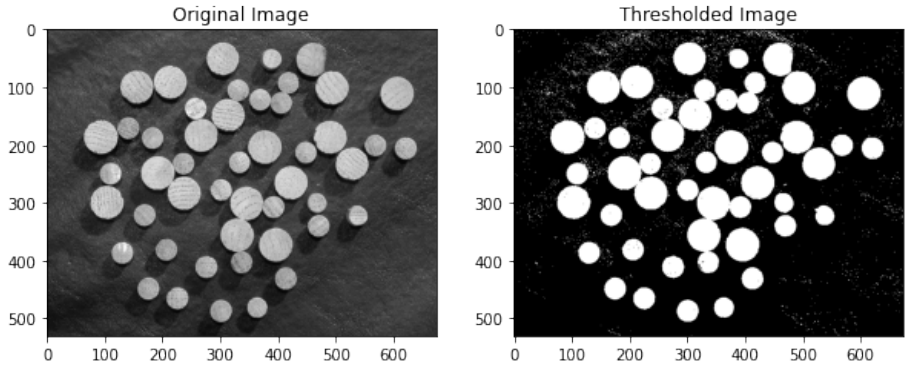
```
<matplotlib.image.AxesImage at 0x2659c839fc8>
```



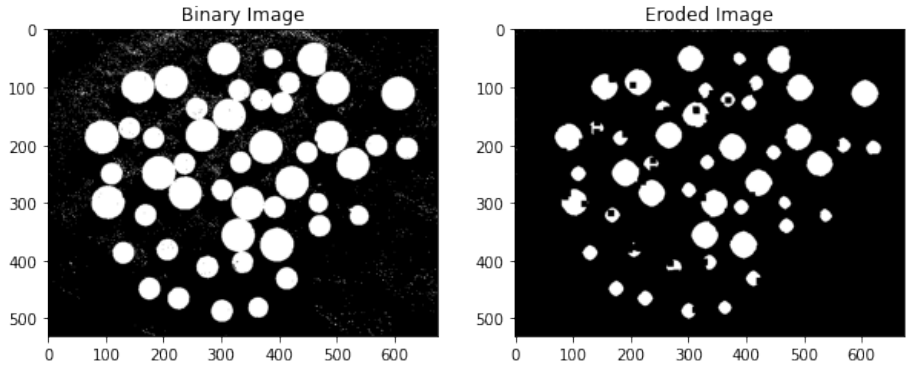

```
image1 = image.copy()
rows = sh[0]
cols = sh[1]
for r in range(rows):
    for c in range(cols):
        if image1[r][c] > 100:
              image1[r][c] = 255
        else:
            image1[r][c]=0
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(image, cmap = 'gray')
plt.title('Original Image')
plt.subplot(1,2,2)
plt.imshow(image1, cmap = 'gray')
plt.title('Thresholded Image')
```

```
Text(0.5, 1.0, 'Thresholded Image')
```



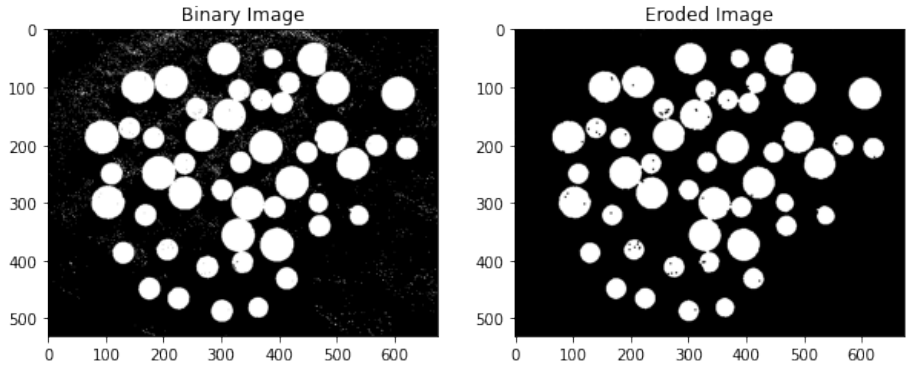

```
# For Erosion
image_err = image1.copy()

sz = 11 # Filter Size
SE = (np.ones([sz,sz]))*255
cent = int ((sz-1)/2)

for r in range(cent, rows):
    for c in range(cent, cols):
        temp = image1[r-cent: r+cent+1,c-cent: c+cent+1]
        if np.array_equal(temp,SE):
            image_err[r][c] = 255
        else:
            image_err[r][c] = 0

plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(image1, cmap='gray')
plt.title('Binary Image')
plt.subplot(1,2,2)
plt.imshow(image_err, cmap='gray')
plt.title('Eroded Image')

Text(0.5, 1.0, 'Eroded Image')
```




```
# Changing Filter Size to 3x3
sz1 = 3
SE = (np.ones([sz1,sz1]))*255
cent = int ((sz1-1)/2)

for r in range(cent, rows):
    for c in range(cent, cols):
        temp = image1[r-cent: r+cent+1,c-cent: c+cent+1]
        if np.array_equal(temp,SE):
            image_err[r][c] = 255
        else:
            image_err[r][c] = 0
```

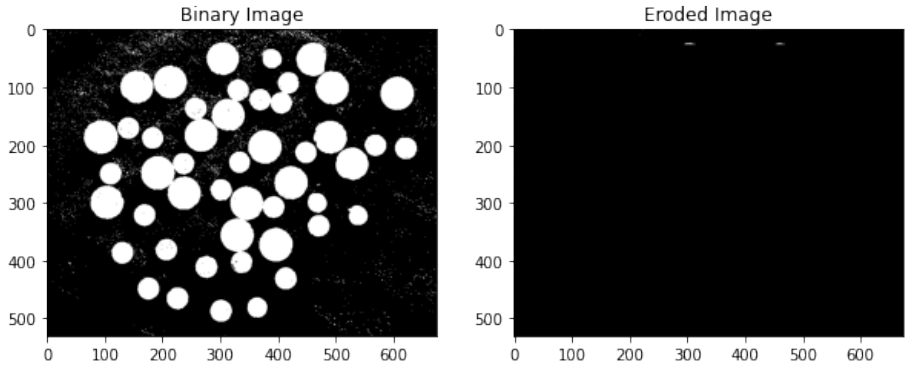
```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(image1, cmap='gray')
plt.title('Binary Image')
plt.subplot(1,2,2)
plt.imshow(image_err, cmap='gray')
plt.title('Eroded Image')
```

```
Text(0.5, 1.0, 'Eroded Image')
```




```
# Changing Filter Size to 55x55 - All circles disappear
sz2 = 55
SE = (np.ones([sz2,sz2]))*255
cent = int ((sz2-1)/2)

for r in range(cent, rows):
    for c in range(cent, cols):
        temp = image1[r-cent: r+cent+1,c-cent: c+cent+1]
        if np.array_equal(temp,SE):
            image_err[r][c] = 255
        else:
            image_err[r][c] = 0

plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(image1, cmap='gray')
plt.title('Binary Image')
plt.subplot(1,2,2)
plt.imshow(image_err, cmap='gray')
plt.title('Eroded Image')
```

```
Text(0.5, 1.0, 'Eroded Image')
```



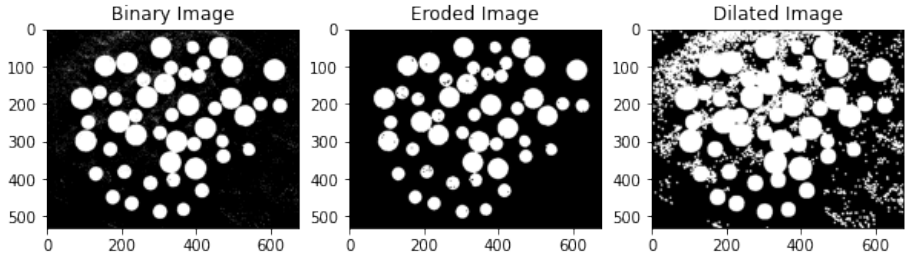
Conclusion

- The given test image is eroded by using square structuring element of size 11x11
- Eroded image shows that disk which are smaller than 11x11 disappear and larger ones reduce in size
- If the size of structuring element is increased to 45, all the discs disappear
- If structuring element is reduced to 3x3, the eroded image shows noisy pixels. Therefore this size is not desirable.

```
image_dilation = image1.copy()

sz = 5 # Filter Size
SE = (np.ones([sz,sz]))*255
cent = int ((sz-1)/2)

for r in range(cent, rows):
    for c in range(cent, cols):
        temp = image1[r-cent: r+cent+1,c-cent: c+cent+1]
        if np.isin(255,temp):
            image_dilation[r][c] = 255
        else:
            image_dilation[r][c] = 0

plt.figure(figsize=(10,10))
plt.subplot(1,3,1)
plt.imshow(image1, cmap='gray')
plt.title('Binary Image')
plt.subplot(1,3,2)
plt.imshow(image_err, cmap='gray')
plt.title('Eroded Image')
plt.subplot(1,3,3)
plt.imshow(image_dilation, cmap='gray')
```

```
plt.title('Dilated Image')
```

```
Text(0.5, 1.0, 'Dilated Image')
```



Conclusion

- The given test image is eroded by using square structuring element of size 11x11
- Dilated image shows increase in size of disc as well as noisy pixel
- To avoid this problem, image is first eroded with structuring element of size 5x5
- This Opened image shows black patches in white disks and therefore the structuring element s increased to 11x11 which reduces the noise to 0 and we get clean discs

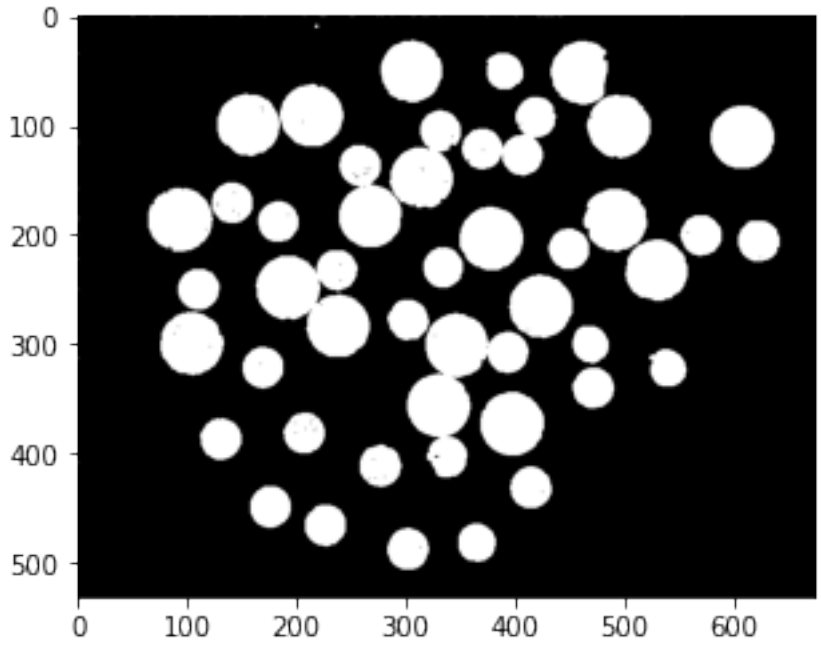
Opening

```
image_dilation1 = image1.copy()
```

```
sz = 3 # Filter Size
SE = (np.ones([sz,sz]))*255
cent = int ((sz-1)/2)
for r in range(cent, rows):
    for c in range(cent, cols):
        temp = image_err[r-cent: r+cent+1,c-cent: c+cent+1]
        if np.isin(255,temp):
            image_dilation1[r][c] = 255
        else:
            image_dilation1[r][c] = 0
```

```
plt.imshow(image_dilation1, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x2659e7ec3c8>
```




```
plt.figure(figsize=(15,15))
plt.subplot(1,3,1)
plt.imshow(image1, cmap='gray')
plt.title('Binary Image')
plt.subplot(1,3,2)
plt.imshow(image_err, cmap='gray')
plt.title('Eroded Image')
plt.subplot(1,3,3)
plt.imshow(image_dilation1, cmap='gray')
plt.title('Image Opening')
```

```
Text(0.5, 1.0, 'Image Opening')
```