

# РОБОТА із ЗОБРАЖЕННЯМИ

## Файл: Image\_06\_001

### Морфологічні перетворення. Операції над множинами

```
## Завантаження пакетів
import numpy as np
import matplotlib.pyplot as plt
import skimage.io as io
plt.rcParams['font.size'] = 10
```

```
# Визначення та друк параметрів зображення
def image_data (image):
    rank      = len(image.shape)
    rows_num  = image.shape[0] ## кількість рядків
    cols_num  = image.shape[1] ## кількість колонок
    chn_num   = 1
    if rank == 3:
        chn_num = image.shape[2] ## кількість каналів
    print('Опис зображення ')
    print('IMAGE RANK',  rank)
    print('IMAGE SHAPE', image.shape)
    print('DATA Type',  image.dtype)
    return rows_num, cols_num, chn_num
```

```
# Відображення обраного зображення
def image1_view (image, image_name):
    fig, ax = plt.subplots(figsize=(4, 4))
    plt.title(image_name)
    plt.imshow(image)
    plt.show()
    return
```

```
# Відображення двох зображень для порівняння
def image2_view (image1, image2, image1_name, image2_name):
    fig, axes = plt.subplots(1, 2, figsize=(8, 4))
    ax = axes.ravel()
    ax[0].set_title(image1_name)
    ax[0].imshow(image1)
    ax[1].set_title(image2_name)
    ax[1].imshow(image2)
    plt.show()
    return
```

```
## Завантаження файлу зображення А множина
path = './IMAGES/'
filename1 = 'A_Image.png'
image_A = io.imread(path+filename1)
```

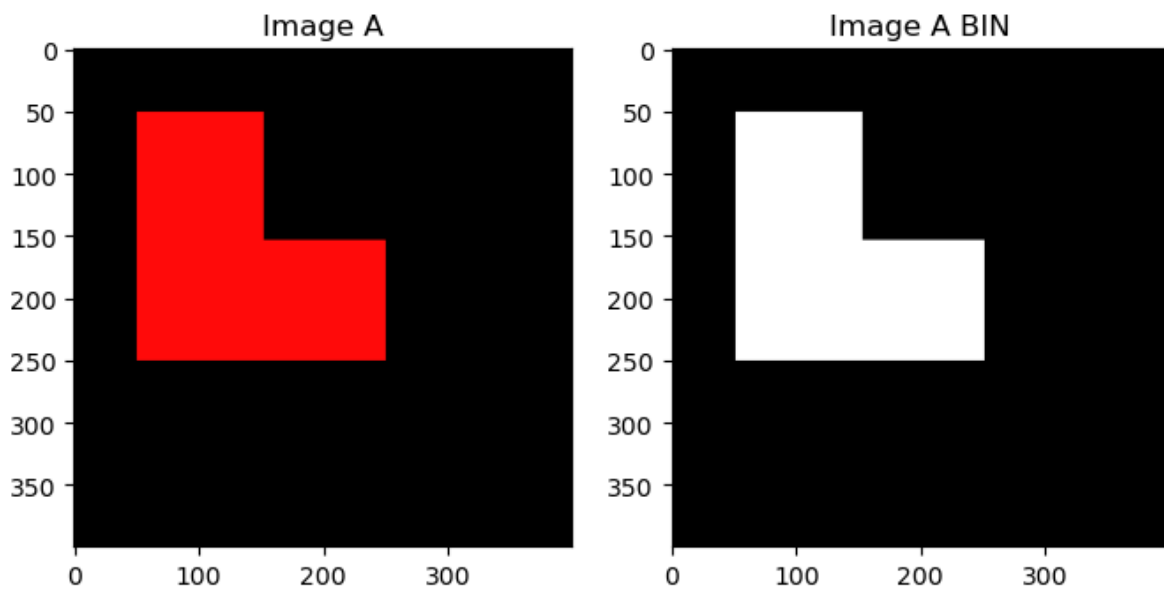
```

rows_num = image_A.shape[0]
cols_num= image_A.shape[1]

# Перетворення до бінарного
bin_image_A = np.full ((rows_num, cols_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (cols_num):
        if image_A[i, j, 0] > 126 : bin_image_A [i, j, :] = [255,255,255]

image2_view (image_A, bin_image_A, 'Image A', 'Image A BIN')

```



```

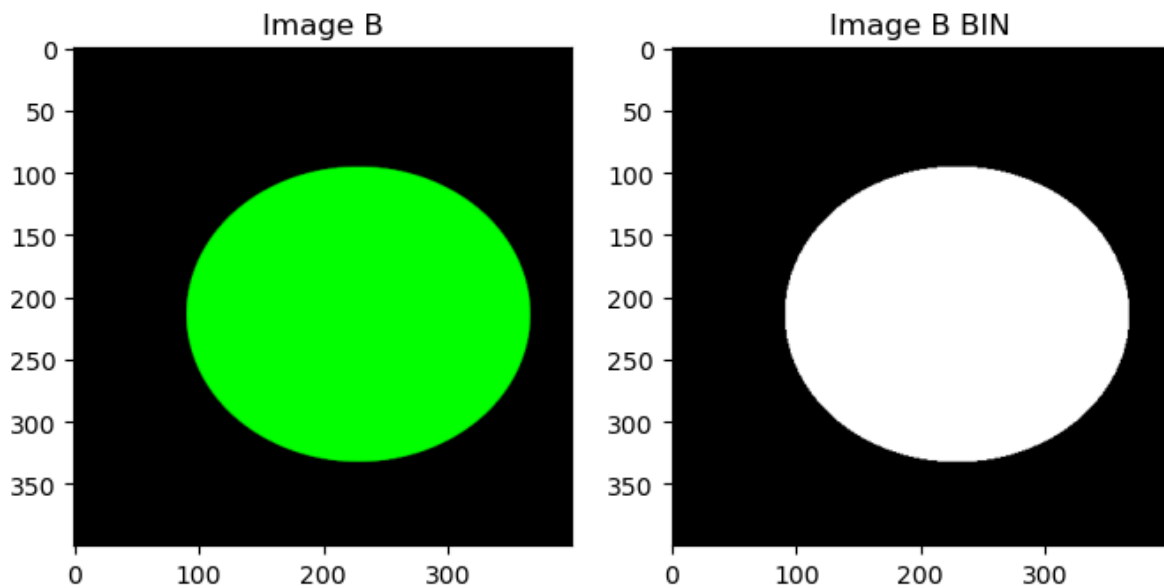
## Завантаження файлу зображення в множина
path = './IMAGES/'
filename1 = 'B_Image.png'
image_B = io.imread(path+filename1)

# перебачаємо що розміри зображень A та B ОДНАКОВІ!

# Перетворення до бінарного
bin_image_B = np.full ((rows_num, cols_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (cols_num):
        if image_B[i, j, 1] > 126 : bin_image_B [i, j, :] = [255,255,255]

image2_view (image_B, bin_image_B, 'Image B', 'Image B BIN')

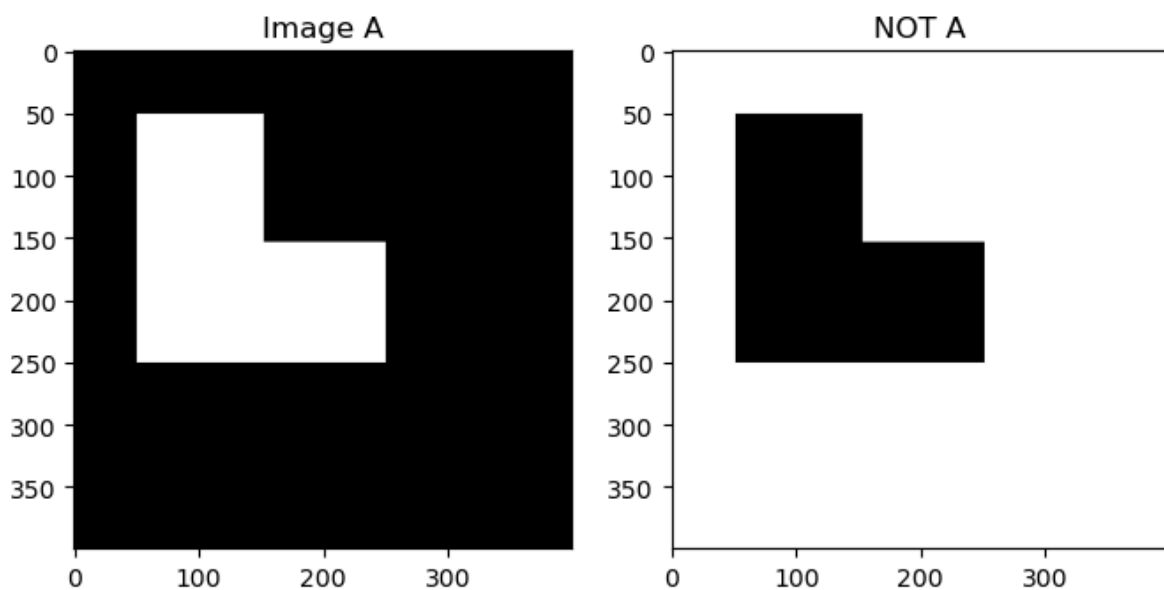
```



### Множинне доповнення (NOT)

```
A_dop = np.full ((rows_num, clms_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (clms_num):
        if bin_image_A[i, j, 0] < 126 :
            A_dop [i,j, :] = [255,255,255]

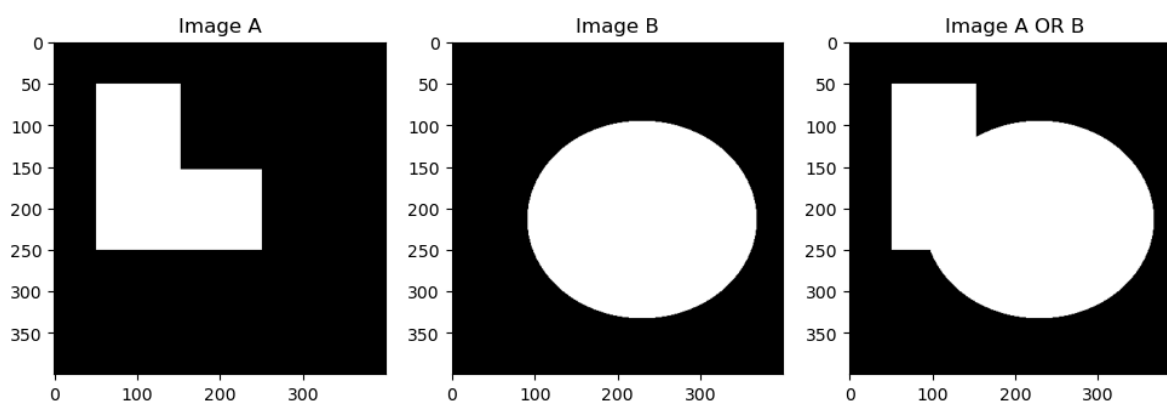
image2_view (bin_image_A, A_dop, 'Image A', 'NOT A')
```



## Множинна операція об'єднання OR

```
A_or_B = np.full ((rows_num, c_lms_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (c_lms_num):
        if bin_image_A[i, j, 0] > 126 or bin_image_B [i, j, 0] > 126:
            A_or_B [i,j, :] = [255,255,255]

fig, axes = plt.subplots(1, 3, figsize=(12, 4))
ax = axes.ravel()
ax[0].imshow(bin_image_A)
ax[0].set_title("Image A")
ax[1].imshow(bin_image_B)
ax[1].set_title("Image B")
ax[2].imshow(A_or_B)
ax[2].set_title("Image A OR B")
plt.show()
```

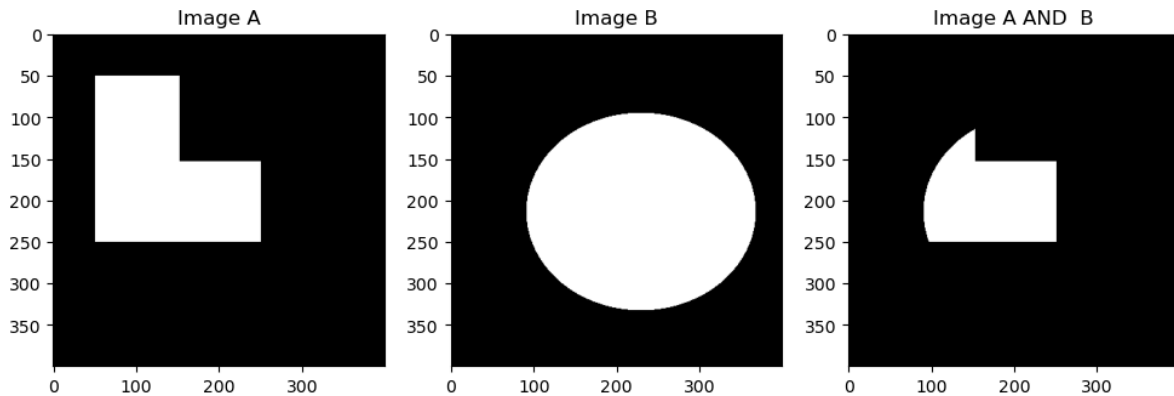


## Множинна операція перетину AND

```
A_and_B = np.full ((rows_num, c_lms_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (c_lms_num):
        if bin_image_A[i, j, 0] > 126 and bin_image_B [i, j, 0] > 126:
            A_and_B [i,j, :] = [255,255,255]

fig, axes = plt.subplots(1, 3, figsize=(12, 4))
ax = axes.ravel()
ax[0].imshow(bin_image_A)
ax[0].set_title("Image A")
ax[1].imshow(bin_image_B)
ax[1].set_title("Image B")
ax[2].imshow(A_and_B)
ax[2].set_title("Image A AND B")

plt.show()
```

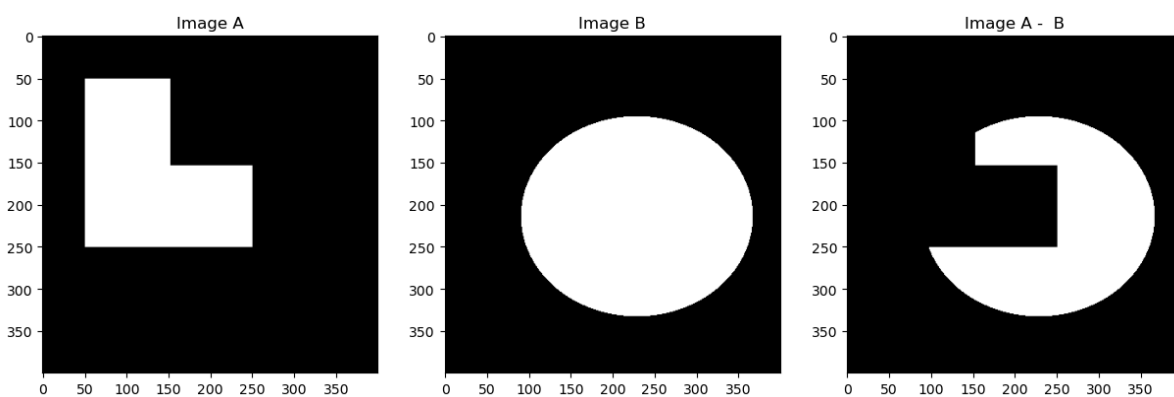


## Множинна операція різниця (A-B)

```
A_min_B = np.full ((rows_num, c_lms_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (c_lms_num):
        if (bin_image_B[i, j, 0] > 126 and bin_image_A [i, j, 0] < 126):
            A_min_B [i,j, :] = [255,255,255]

fig, axes = plt.subplots(1, 3, figsize=(15, 5))
ax = axes.ravel()
ax[0].imshow(bin_image_A)
ax[0].set_title("Image A")
ax[1].imshow(bin_image_B)
ax[1].set_title("Image B")
ax[2].imshow(A_min_B)
ax[2].set_title("Image A - B")

plt.show()
```



## Множинна операція симетрична різниця XOR

```
A_xor_B = np.full ((rows_num, c_lms_num, 3), (0, 0, 0) , dtype=np.uint8)
for i in range (rows_num):
    for j in range (c_lms_num):
        if (bin_image_A[i, j, 0] > 126 and bin_image_B [i, j, 0] < 126) or
(bin_image_A[i, j, 0] < 126 and bin_image_B [i, j, 0] > 126):
            A_xor_B [i,j, :] = [255,255,255]
```

```

fig, axes = plt.subplots(1, 3, figsize=(15, 5))
ax = axes.ravel()
ax[0].imshow(bin_image_A)
ax[0].set_title("Image A")
ax[1].imshow(bin_image_B)
ax[1].set_title("Image B")
ax[2].imshow(A_xor_B)
ax[2].set_title("Image A XOR B")

plt.show()

```

