

## THỰC HÀNH XỬ LÝ ẢNH SỐ - BÀI 3

Nội dung chính của bài thực hành thứ ba sẽ tập trung vào các phép toán hình thái trên không gian ảnh và ứng dụng của chúng trong tăng cường chất lượng và trích chọn thông tin.

### LÝ THUYẾT VÀ Ý NGHĨA CÁC THAO TÁC HÌNH THÁI TRÊN KHÔNG GIAN ẢNH

Các thao tác hình thái thường áp dụng cho ảnh nhị phân (binary), và một số trường hợp là ảnh đa cấp xám (grey scale). Đặc điểm chung của các phép hình thái là xử lý trên các vùng/đối tượng trong ảnh (các nhóm điểm ảnh có cùng tương quan về màu sắc một cách tương đối), không quan tâm đến giá trị cường độ hay giá trị màu sắc thực sự của điểm ảnh.

#### CÁC PHÉP CO (ERODE) VÀ GIÃN (DILATE)

ERODE (Xói mòn – Tùy theo chúng ta định nghĩa object hoặc vùng trên ảnh là màu trắng hay màu đen):

- Loại bỏ các điểm ảnh trắng (sáng) bị cô lập;
- từ đó loại bỏ các điểm ảnh nhiều cũng như loại bỏ các điểm gai (điểm nhiễu) trên đường biên của một vùng/đối tượng trong ảnh – đường biên trở nên mịn hơn;
- Loại bỏ lớp viền (cạnh) của đối tượng giúp đối tượng trở nên nhỏ hơn và đặt những pixel viền đó trở thành lớp nền của đối tượng.

DILATE (Giãn nở ảnh):

- Với những hình ảnh bị đứt nét có thể giúp nối liền ảnh lại
- Với những pixel nhiễu xung quanh đối tượng sẽ trở thành viền của đối tượng
- Giúp nổi bật đối tượng trong ảnh hơn

Công thức toán của các phép Erode và Dilate:

- **dilation:**  $\text{dilate}(f, s) = \theta(c, 1);$
- **erosion:**  $\text{erode}(f, s) = \theta(c, S);$

Trong đó:

$$c = f \otimes s$$

và

$$\theta(f, t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases}$$

Trong công thức trên ta có sự xuất hiện của mặt nạ (bộ lọc)  $s$ , ở đây  $s$  được gọi là các phần tử cấu trúc của ảnh. Các bộ lọc này cho phép xác định đâu là điểm lân cận với điểm đang xét và cần xử lý.

#### CÁC PHÉP MỞ (OPENING) VÀ ĐÓNG (CLOSING)

Thực hiện phép co (Erosion) sau đó thực hiện phép giãn (Dilation) lên kết quả từ phép co, ta thu được phép mở - OPENING.

Thực hiện phép giãn (Dilation) sau đó thực hiện phép co (Erosion) lên kết quả từ phép giãn, ta thu được phép đóng (CLOSING).

## THỰC HÀNH VỚI SKIMAGE

```
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread, imshow
from skimage.draw import disk
from skimage.morphology import (erosion, dilation, closing, opening,
area_closing, area_opening)
from skimage.color import rgb2gray
circle_image = np.zeros((25, 40))
circle_image[disk((12, 12), 8)] = 1
circle_image[disk((12, 28), 8)] = 1
for x in range(20):
    circle_image[np.random.randint(25), np.random.randint(40)] = 1
imshow(circle_image);
fig, ax = plt.subplots(1,2, figsize=(15,5))
ax[0].imshow(erosion(circle_image, element), cmap='gray');
ax[0].set_title('Eroded Image')
ax[1].imshow(dilation(circle_image, element), cmap='gray')
ax[1].set_title('Dilated Image')
def multi_dil(im, num, element=element):
    for i in range(num):
        im = dilation(im, element)
    return im
def multi_ero(im, num, element=element):
    for i in range(num):
        im = erosion(im, element)
    return im
fig, ax = plt.subplots(1,2, figsize=(15,5))
ax[0].imshow(multi_ero(circle_image, 2, element), cmap='gray')
ax[0].set_title('Multi-Eroded Image')
ax[1].imshow(multi_dil(circle_image, 2, element), cmap='gray')
ax[1].set_title('Multi-Dilated Image')
fig, ax = plt.subplots(1,2, figsize=(15,5))
ax[0].imshow(opening(circle_image, element), cmap='gray');
ax[0].set_title('Opened Image')
ax[1].imshow(closing(circle_image, element), cmap='gray')
ax[1].set_title('Closed Image')
```

## THỰC HÀNH VỚI OPENCV

Dưới đây chúng ta sử dụng thư viện OpenCV để thực hiện các thao tác hình thái trên ảnh nhằm tách các đối tượng. Trong ảnh minh họa ta có một số đồng xu, chúng ta sẽ thực hiện các thao tác hình thái để tách được hình dạng các đồng xu đó ra khỏi nền.

Các thao tác co - giãn

```
import cv2
import numpy as np

# Reading the input image
img = cv2.imread('D:\\Teach_n_Train\\ImageProcessing\\code\\material\\test3.png', 0)

# Taking a matrix of size 5 as the kernel
kernel = np.ones((5,5), np.uint8)

img_erosion = cv2.erode(img, kernel, iterations=1)
img_dilation = cv2.dilate(img, kernel, iterations=1)

cv2.imshow('Input', img)
```

```
cv2.imshow('Erosion', img_erosion)
cv2.imshow('Dilation', img_dilation)

cv2.waitKey(0)
if k == 27: # wait for ESC key to exit
    cv2.destroyAllWindows()
```

## Các thao tác đóng – mở

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

# Image operation using thresholding
img = cv2.imread('D:\\Teach_n_Train\\ImageProcessing\\code\\material\\test3.png')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

ret, thresh = cv2.threshold(gray, 0, 255,
                             cv2.THRESH_BINARY_INV +
                             cv2.THRESH_OTSU)
cv2.imshow('Org image', thresh)

k = cv2.waitKey(0)
if k == 27: # wait for ESC key to exit
    cv2.destroyAllWindows()
kernel = np.ones((5, 5), np.uint8)

closing = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE,
                           kernel, iterations = 2)

cv2.imshow('Closed image', closing)
k = cv2.waitKey(0)
if k == 27: # wait for ESC key to exit
    cv2.destroyAllWindows()
```