**a. Hiển thị ảnh fingerprint.tif**

import numpy as np

# Đọc ảnh fingerprint.tif

img = cv2.imread('fingerprint.tif', cv2.IMREAD\_GRAYSCALE)

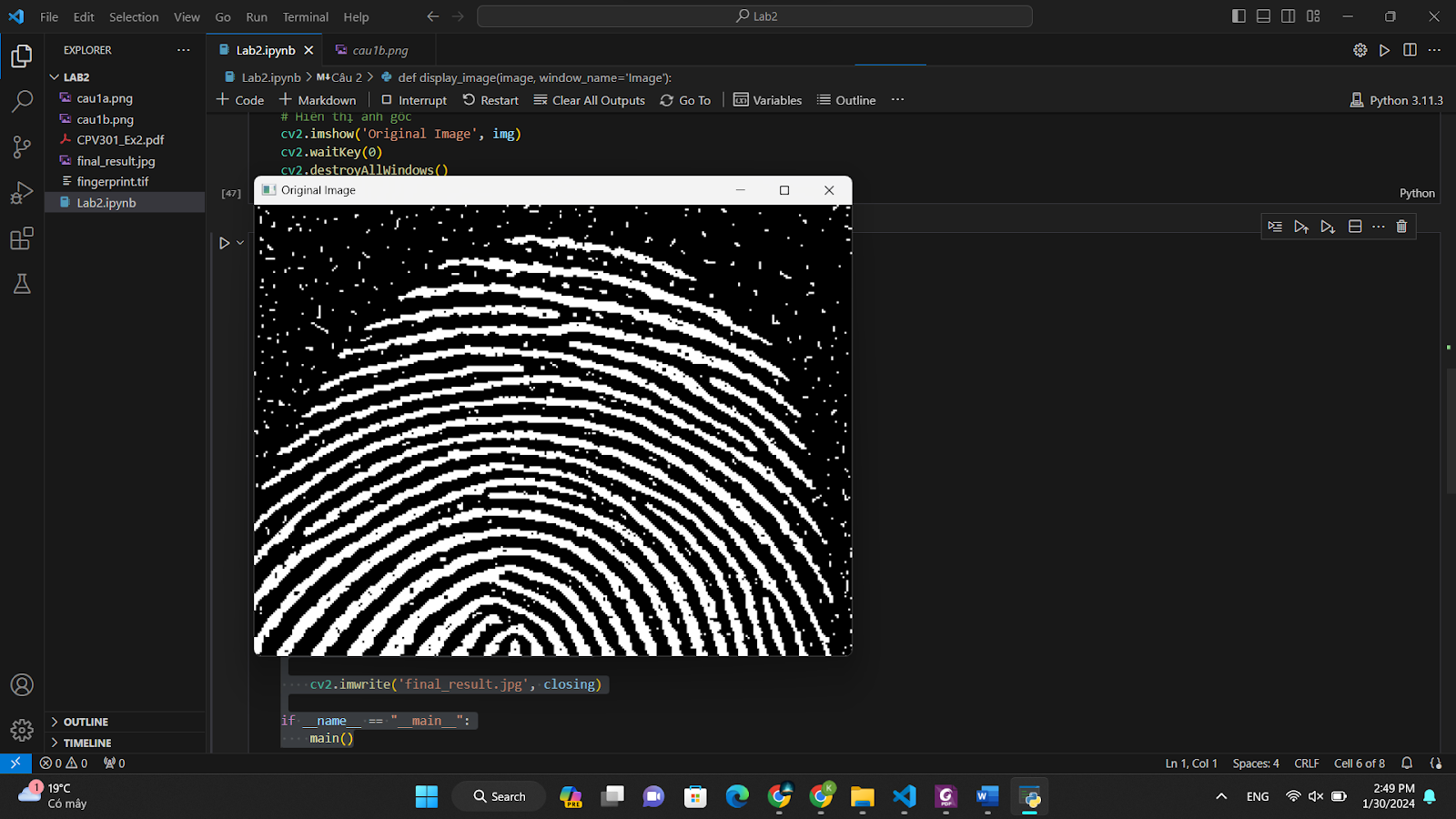
# Hiển thị ảnh gốc

cv2.imshow('Original Image', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

KẾT QUẢ

  
**b. Thực hiện các phép toán co, giãn, đóng, mở trên ảnh gốc để loại bỏ nhiễu.**

def display\_image(image, window\_name='Image'):

    cv2.imshow(window\_name, image)

    cv2.waitKey(0)

    cv2.destroyAllWindows()

def main():

    # Đọc ảnh fingerprint

    img = cv2.imread('fingerprint.tif', cv2.IMREAD\_GRAYSCALE)

    # Thực hiện phép toán co

    kernel = np.ones((5, 5), np.uint8)

    erosion = cv2.erode(img, kernel, iterations=1)

    display\_image(erosion, 'Erosion')

    # Thực hiện phép toán giãn

    dilation = cv2.dilate(erosion, kernel, iterations=1)

    display\_image(dilation, 'Dilation')

    # Thực hiện phép toán mở

    opening = cv2.morphologyEx(dilation, cv2.MORPH\_OPEN, kernel)

    display\_image(opening, 'Opening')

    # Thực hiện phép toán đóng

    closing = cv2.morphologyEx(opening, cv2.MORPH\_CLOSE, kernel)

    display\_image(closing, 'Closing')

    # Hiển thị kết quả cuối cùng

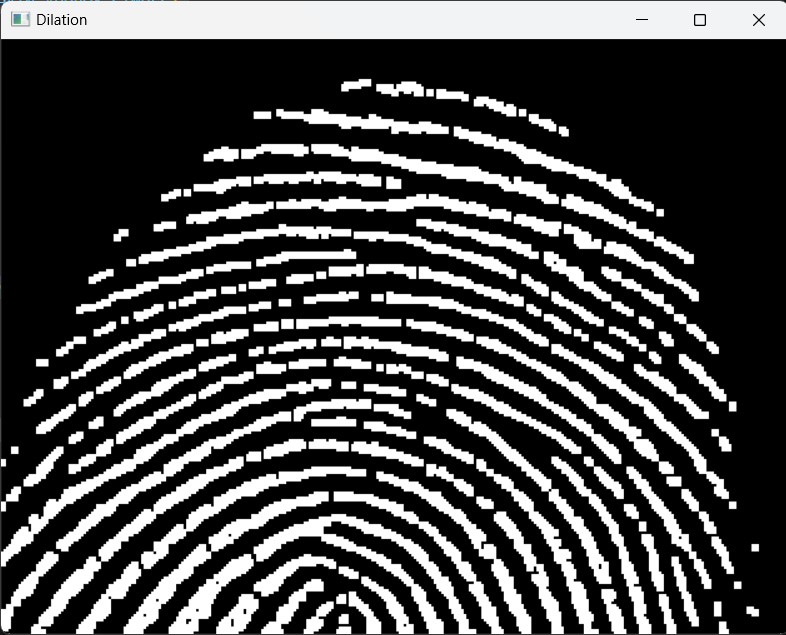
    display\_image(closing, 'Final Result')

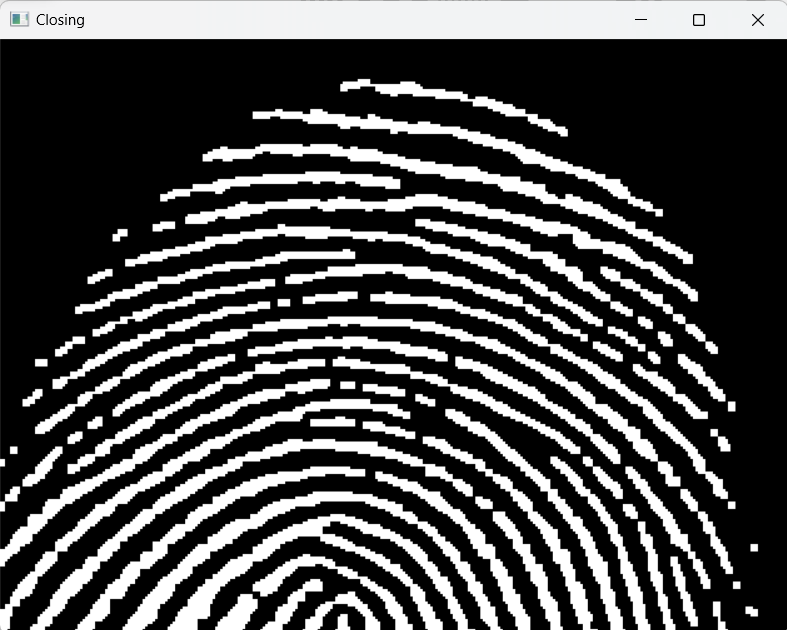
    cv2.imwrite('final\_result.jpg', closing)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**KẾT QUẢ**



**c. Đặc điểm của vân tay cho nhận dạng là gì? Code python để đánh dấu trên ảnh các điểm đặc trưng, ảnh thu được từ câu b.**

Đặc điểm của vân tay:

- Số lỗ mà vân tay có

- Các đường gấp khúc (ridge) và các điểm nối giữa chúng

- Độ dày của các dãy đường gấp khúc

def mark\_minutiae(image):

    # Tìm contours trên ảnh đã xử lý

    contours,\_=cv2.findContours(image,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE)

    # Tạo một bản sao của ảnh để vẽ các điểm đặc trưng

    marked\_img = cv2.cvtColor(image, cv2.COLOR\_GRAY2BGR)

    # Duyệt qua tất cả các contours

    for contour in contours:

        for point in contour:

            # Lấy tọa độ của điểm

            x, y = point[0]

            # Vẽ một hình tròn tại điểm đặc trưng

            cv2.circle(marked\_img, (x, y), 3, (0, 0, 255), -1)  # Màu đỏ

    return marked\_img

def main():

    # Đọc ảnh từ phần cuối của đoạn mã bạn đã cung cấp

    closing = cv2.imread('final\_result.jpg', cv2.IMREAD\_GRAYSCALE)

    # Đánh dấu các điểm đặc trưng trên ảnh đã xử lý

    marked\_image = mark\_minutiae(closing)

    # Hiển thị ảnh đã đánh dấu các điểm đặc trưng

    display\_image(marked\_image, 'Minutiae Marked Image')

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**KẾT QUẢ**

