1. Lập trình đọc ảnh từ webcam, khi ảnh ổn định ấn S thì lưu ảnh vào thư mục trong máy (có thể chọn thư mục, đặt tên file hoặc lấy tên file theo thời gian).
2. Ấn phím F thì lọc tách cạnh (edge) trước khi lưu ảnh.

import cv2

import os

from datetime import datetime

def save\_image(img, folder, filename=None):

    if not os.path.exists(folder):

        os.makedirs(folder)

    if filename is None:

        filename = datetime.now().strftime("%Y%m%d\_%H%M%S") + ".jpg"

    cv2.imwrite(os.path.join(folder, filename), img)

    print("Saved image as", filename)

def edge\_detection(img):

    gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

    edges = cv2.Canny(gray, 100, 200)

    return cv2.cvtColor(edges, cv2.COLOR\_GRAY2BGR)

def main():

    cap = cv2.VideoCapture(0)

    while True:

        ret, frame = cap.read()

        cv2.imshow('Webcam', frame)

        key = cv2.waitKey(1)

        if key == ord('s') or key == ord('S'):

            folder = input("Enter folder path to save the image: ")

            filename = input("Enter filename (leave blank to use timestamp): ")

            save\_image(frame, folder, filename)

        elif key == ord('f') or key == ord('F'):

            folder = input("Enter folder path to save the edge-detected image: ")

            filename = input("Enter filename (leave blank to use timestamp): ")

            edge\_img = edge\_detection(frame)

            save\_image(edge\_img, folder, filename)

        elif key == 27:  # 27 is the ASCII value for the Escape key

            break

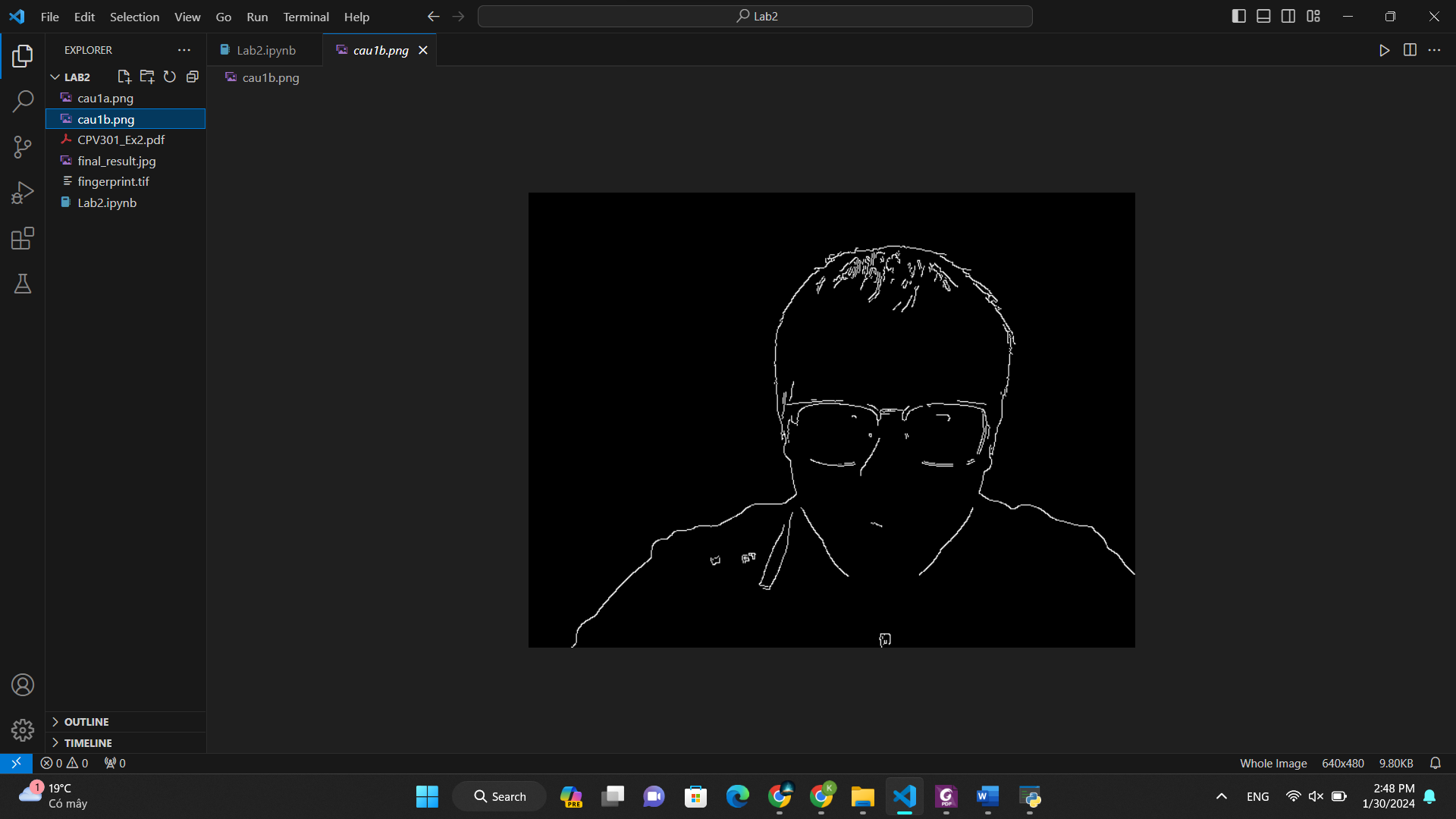
    cap.release()

    cv2.destroyAllWindows()

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

    main()

KẾT QUẢ

1. Đưa điện thoại có chữ trước webcam và chụp sao cho phần hình điện   
   thoại là hình tứ giác nhỏ. Thực hiện biến đổi để ảnh điện thoại tràn ra   
   hình chữ nhật của toàn màn hình.

import cv2

import numpy as np

def capture\_image():

    # Mở webcam

    cap = cv2.VideoCapture(0)

    while True:

        # Đọc khung hình từ webcam

        ret, frame = cap.read()

        # Hiển thị khung hình

        cv2.imshow('Press S to capture', frame)

        # Chờ phím được nhấn

        key = cv2.waitKey(1)

        # Nếu phím 's' được nhấn, chụp ảnh và lưu lại

        if key == ord('s'):

            cv2.imwrite('captured\_image.jpg', frame)

            print("Image captured!")

            break

    # Đóng webcam

    cap.release()

    cv2.destroyAllWindows()

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

    capture\_image()

# Biến global để lưu trữ các điểm được chọn

points = []

def mouse\_click(event, x, y, flags, param):

    # Xác định sự kiện nhấp chuột

    if event == cv2.EVENT\_LBUTTONDOWN:

        # Thêm tọa độ vào danh sách points

        points.append((x, y))

        # Vẽ điểm trên ảnh

        cv2.circle(image, (x, y), 3, (0, 255, 0), -1)

        cv2.imshow('Image', image)

# Load ảnh

image = cv2.imread('captured\_image.jpg')

# Sao chép ảnh để tránh thay đổi ảnh gốc

clone = image.copy()

# Tạo cửa sổ và kết nối với hàm xử lý sự kiện

cv2.namedWindow('Image')

cv2.setMouseCallback('Image', mouse\_click)

# Hiển thị ảnh và chờ sự kiện

cv2.imshow('Image', image)

cv2.waitKey(0)

cv2.destroyAllWindows()

# Lưu toạ độ vào file txt

with open('coordinates.txt', 'w') as f:

    for point in points:

        f.write(f"{point[0]}, {point[1]}\n")

# Đọc tọa độ từ file txt

with open('coordinates.txt', 'r') as f:

    points = [tuple(map(int, line.strip().split(', '))) for line in f]

# Tính toạ độ của hình chữ nhật bao quanh 4 điểm

x\_sorted = sorted(points, key=lambda x: x[0])

y\_sorted = sorted(points, key=lambda x: x[1])

left = x\_sorted[0][0]

right = x\_sorted[-1][0]

top = y\_sorted[0][1]

bottom = y\_sorted[-1][1]

# Tạo ma trận biến đổi từ hình tứ giác thành hình chữ nhật

input\_pts = np.float32(points)

output\_pts = np.float32([[left, top], [right, top], [right, bottom], [left, bottom]])

# Tính ma trận biến đổi và biến đổi hình ảnh

transform\_matrix = cv2.getPerspectiveTransform(input\_pts, output\_pts)

image = cv2.imread('captured\_image.jpg')

warped\_image = cv2.warpPerspective(image, transform\_matrix, (image.shape[1], image.shape[0]))

# Hiển thị hình ảnh biến đổi

cv2.imshow('Warped Image', warped\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

KẾT QUẢ :   
