import cv2

import numpy as np

import matplotlib.pyplot as plt

image = cv2.imread('image.jpg')

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

blurred = cv2.GaussianBlur(gray, (5, 5), 0)

edged = cv2.Canny(blurred, 30, 150)

dilated = cv2.dilate(edged, None, iterations=2)

eroded = cv2.erode(dilated, None, iterations=1)

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

markers = np.zeros\_like(gray)

cv2.drawContours(markers, contours, -1, (255), -1)

dist\_transform = cv2.distanceTransform(cv2.bitwise\_not(edged), cv2.DIST\_L2, 5)

\_, sure\_fg = cv2.threshold(dist\_transform, 0.7 \* dist\_transform.max(), 255, 0)

sure\_fg = np.uint8(sure\_fg)

sure\_bg = cv2.dilate(sure\_fg, None, iterations=3)

unknown = cv2.subtract(sure\_bg, sure\_fg)

\_, markers = cv2.connectedComponents(sure\_fg)

markers = markers + 1

markers[unknown == 255] = 0

cv2.watershed(image, markers)

image[markers == -1] = [0, 0, 255]

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

plt.title('Watershed Segmentation')

plt.axis('off')

plt.subplot(2, 2, 2)

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

plt.title('Grayscale Image')

plt.axis('off')

plt.subplot(2, 2, 3)

plt.imshow(edged, cmap='gray')

plt.title('Edge Detection')

plt.axis('off')

plt.subplot(2, 2, 4)

plt.imshow(cv2.cvtColor(cv2.bitwise\_not(unknown), cv2.COLOR\_BGR2RGB))

plt.title('Unknown Region')

plt.axis('off')

plt.tight\_layout()

plt.show()