

# Binary image processing –detecting blobs, contours, and central axis

## CODE

```
In [*]: import cv2 as cv
import numpy as np

img1 = input("Enter an Image: ")
img = cv.imread(img1)

kernel = np.ones((3,3), np.uint8)
kernel1 = np.ones((5,5), np.uint8)
kernel2 = np.ones((7,7), np.uint8)
kernel3 = np.ones((9,9), np.uint8)
kernel4 = np.ones((13,13), np.uint8)

for i in range(2):
    img1 = cv.erode(img, kernel1,i)
    final = cv.dilate(img1, kernel,i)

cv.imwrite("wall1-blobs.png", final)

for i in range(0, img.shape[1]):
    final[0, i] = 255
    final[img.shape[0] - 1, i] = 255
for i in range(0, img.shape[0]):
    final[i, 0] = 255
    final[i, img.shape[1] - 1] = 255

final1 = cv.cvtColor(final, cv.COLOR_BGR2GRAY)
ret, thresh = cv.threshold(final1, 127,255,0)
contours, hierarchy= cv.findContours(thresh, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)

for i in range(len(contours)):
    color = (np.random.randint(0,256), np.random.randint(0,256), np.random.randint(0,256))
    cv.drawContours(final, contours, i, color, 3)

cv.imwrite("wall1-contours.png", final)

img_cont = np.zeros(final1.shape, dtype = np.uint8)
img_cont.fill(255)

cont = []
for i in range (len(contours)):
    cont.append(cv.contourArea(contours[i]))
cont = np.array(cont)
np.where((cont > 1000))
for i in range (np.shape(np.where(cont> 1000))[1]):
    cv.drawContours(img_cont, contours,np.where((cont > 1000))[0][i] , (0,0,0), 1)

img_cont = cv.bitwise_not(img_cont)
img_cont = cv.morphologyEx(img_cont,cv.MORPH_CLOSE,kernel4)

kernel5 = cv.getStructuringElement(cv.MORPH_CROSS,(3,3))
# Create an empty output image to hold values
thin = np.zeros(final1.shape,dtype='uint8')

# Loop until erosion leads to an empty set
while (cv.countNonZero(img_cont)!=0):
    # Erosion
    erode = cv.erode(img_cont,kernel5)
    # Opening on eroded image
    opening = cv.morphologyEx(erode,cv.MORPH_OPEN,kernel5)
    # Subtract these two
    subset = erode - opening
    # Union of all previous sets
    thin = cv.bitwise_or(subset,thin)
    # Set the eroded image for next iteration
    img_cont = erode.copy()

thin = cv.bitwise_not(thin)

#img2 = cv.bitwise_not(img2)'''
cv.imshow("Original Image", img)
cv.imshow("Final Output", final)
cv.imshow("Contour", thin)
cv.waitKey(0)
cv.destroyAllWindows()
```

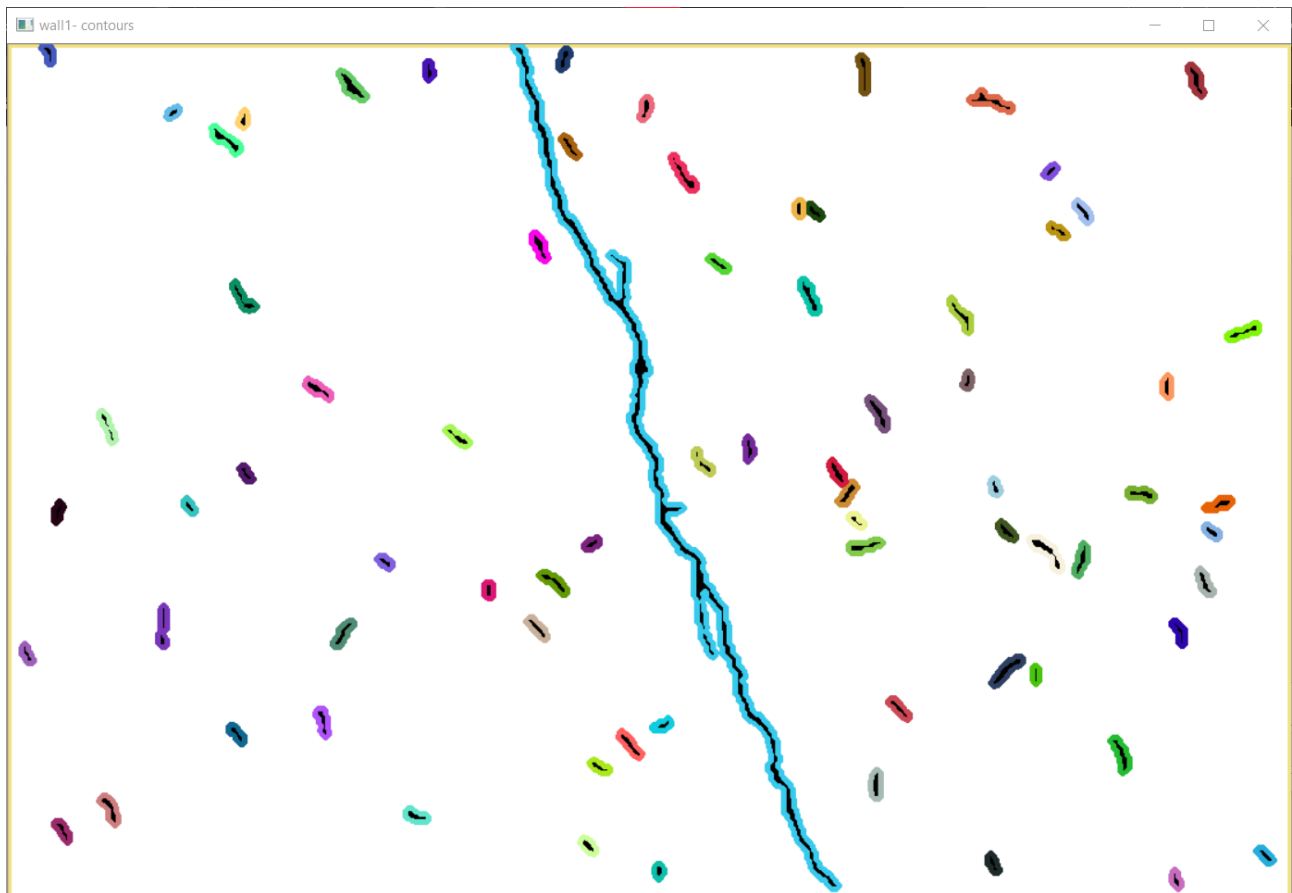
RESULTS:

WALL1.PNG

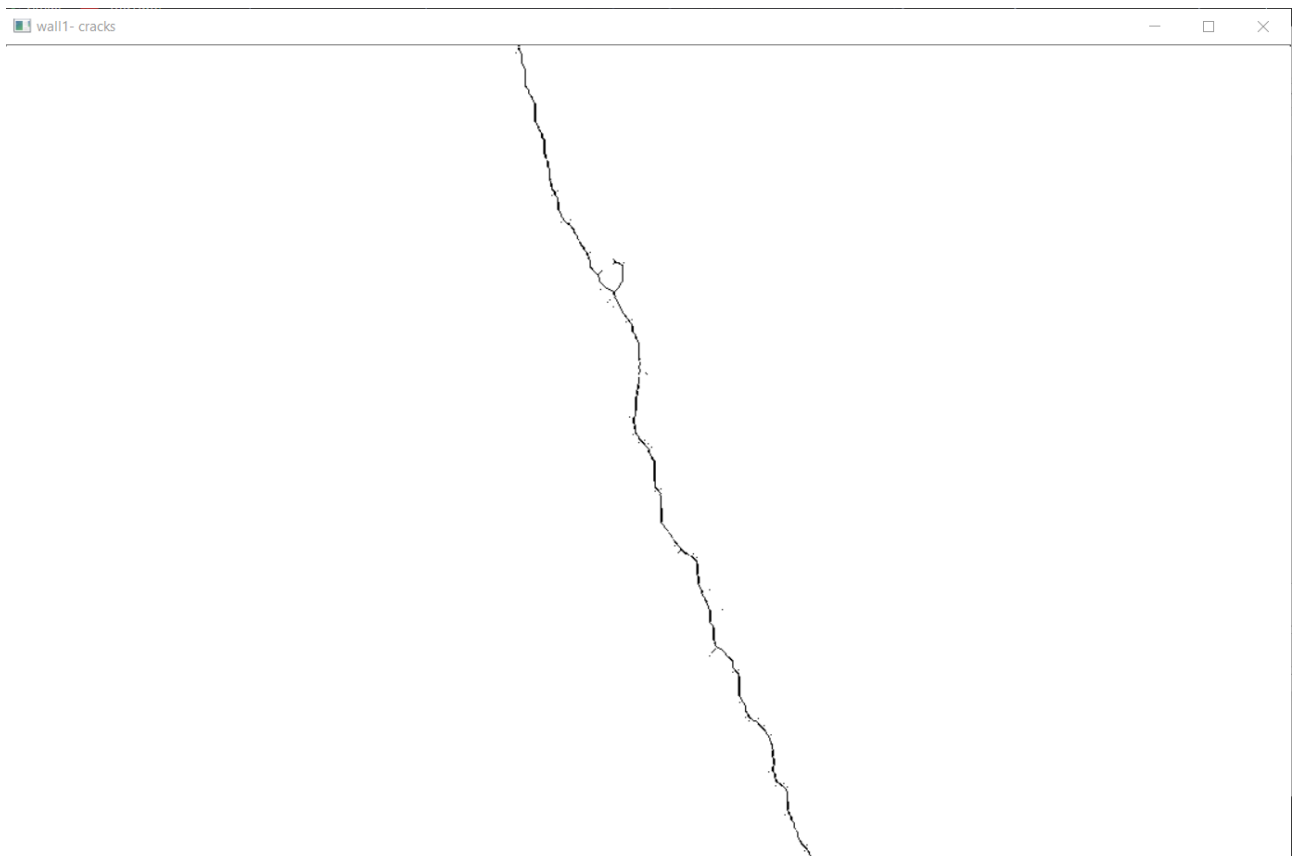
Wall1-blobs.png



Wall1- contours



Wall1- cracks

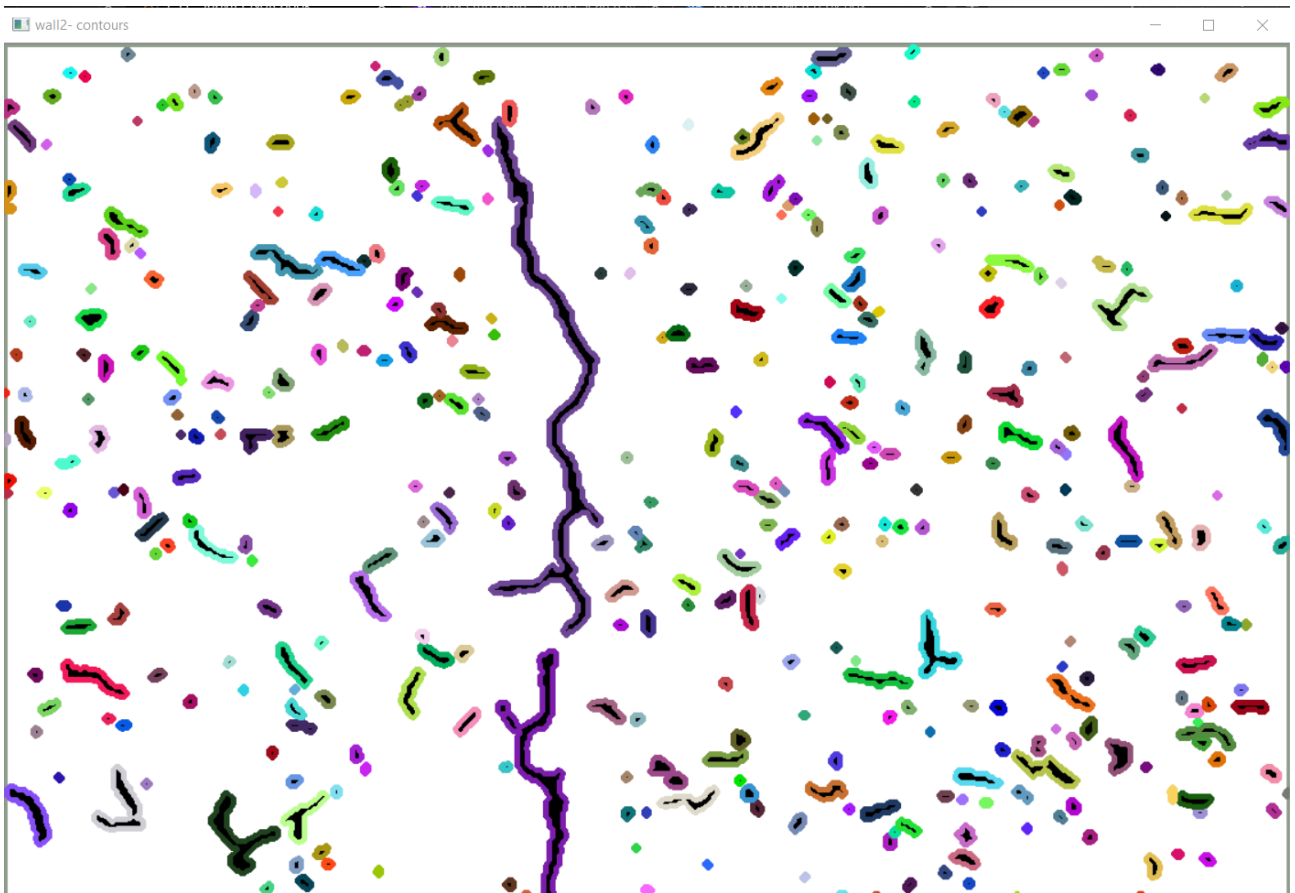


WALL2.PNG

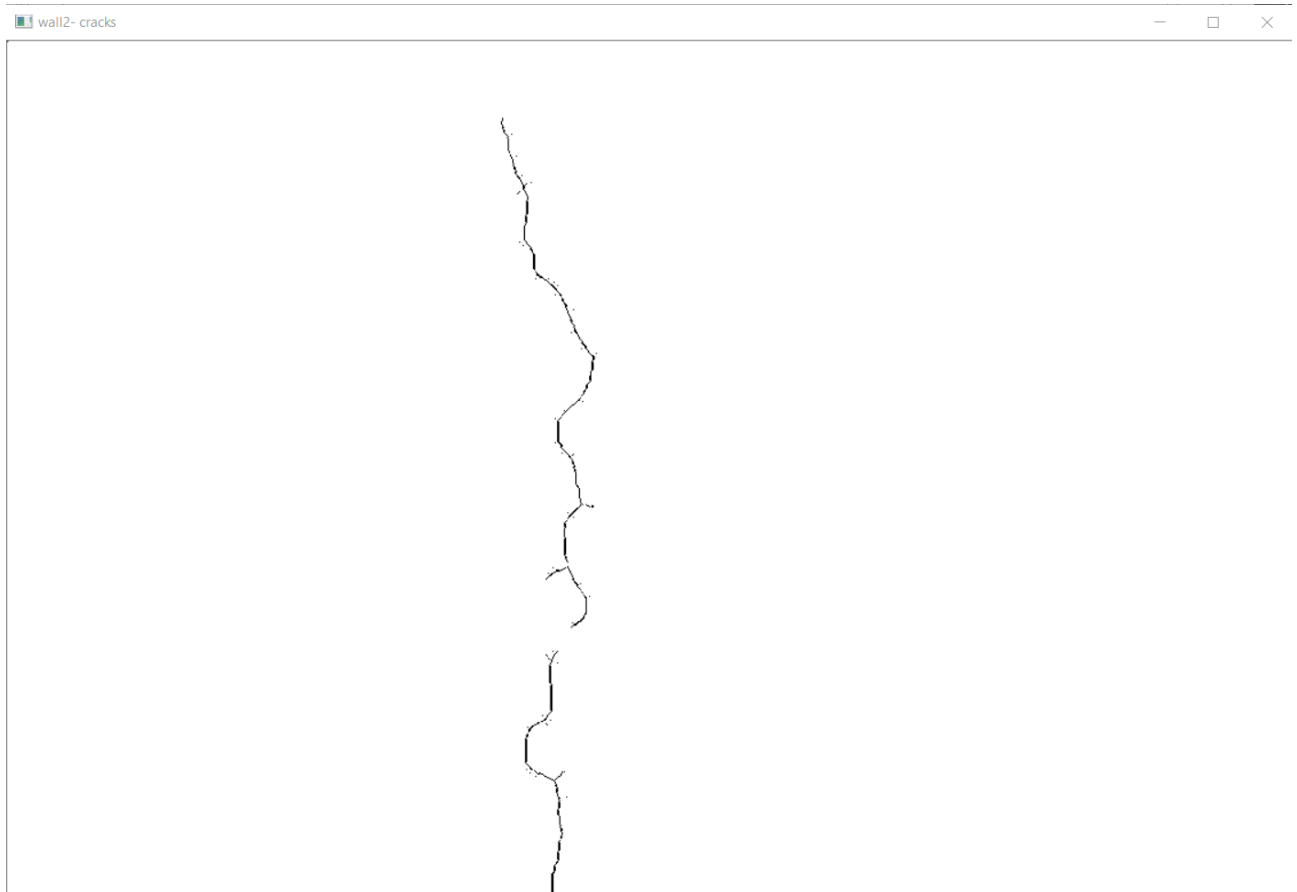
Wall2-blobs



Wall2- contours



## Wall2- cracks



Operating System: Windows 10

IDE: Jupyter Notebook

Number of Hours Spent: 8 Hours