

4. Контури

1. Знаходження контурів на зображені

КОД:

```
import cv2

import numpy as np


img = cv2.imread("/home/rodion/yuliia0/aboba/cv/cont/cool.jpg")

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

gray = cv2.blur(gray, (3,3))

cv2.imshow('cool', img)


canny= cv2.Canny(gray, 60, 60 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)


drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

for i in range(len(contours)):

cv2.drawContours(drawing, contours, i, (250,50,190), 2, cv2.LINE_8,
hierarchy, 0)

cv2.imshow('Contours 1', drawing)


canny= cv2.Canny(gray, 20, 20 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

```
drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

for i in range(len(contours)):

cv2.drawContours(drawing, contours, i, (250,50,190), 2, cv2.LINE_8,
hierarchy, 0)

cv2.imshow('Contours 2', drawing)

canny= cv2.Canny(gray, 5, 5 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

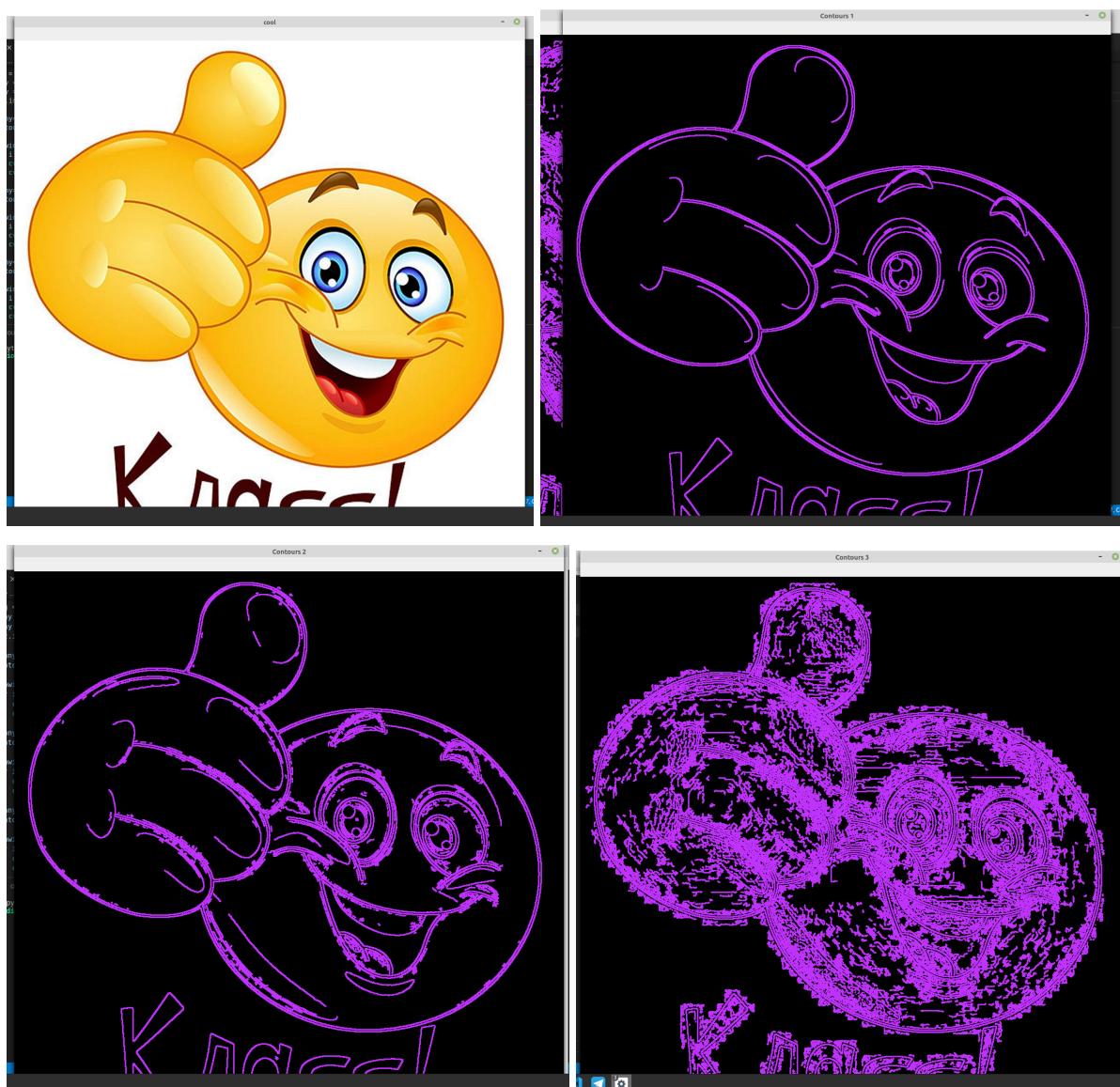
for i in range(len(contours)):

cv2.drawContours(drawing, contours, i, (250,50,190), 2, cv2.LINE_8,
hierarchy, 0)

cv2.imshow('Contours 3', drawing)

cv2.waitKey(0)
```

результат:



2. Опукла оболонка

КОД:

```
import cv2
import numpy as np

img = cv2.imread("/home/rodion/yuliia0/aboba/cv/cont/cool.jpg")

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

gray = cv2.blur(gray, (3,3))

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

```
canny= cv2.Canny(gray, 60, 60 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

hull_list = []

drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

for i in range(len(contours)):

hull = cv2.convexHull(contours[i])

hull_list.append(hull)

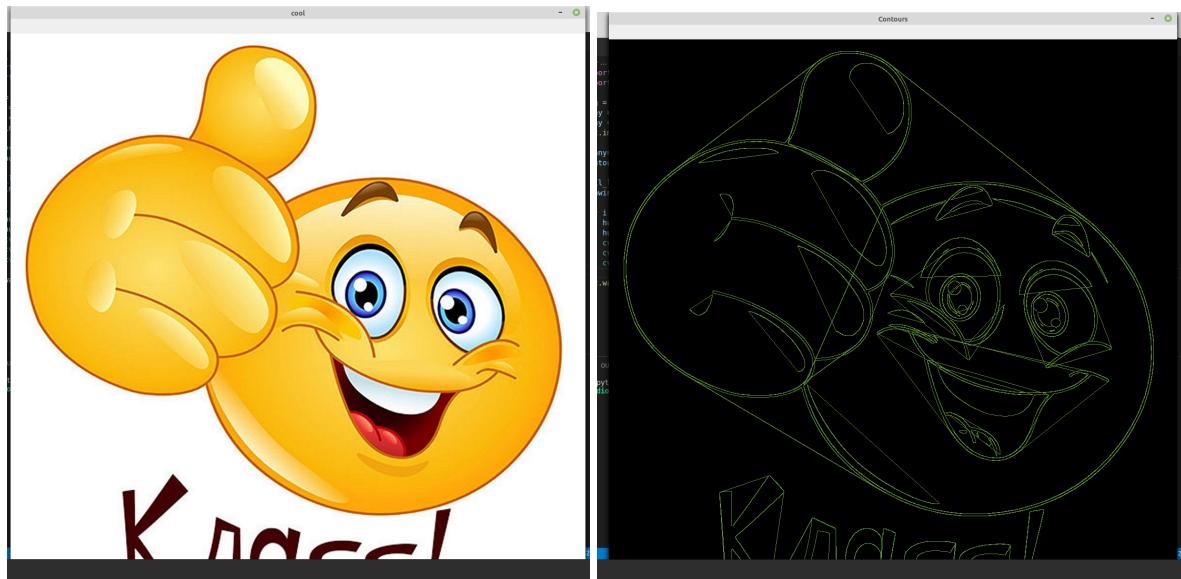
cv2.drawContours(drawing, contours, i, (50,150,100))

cv2.drawContours(drawing, hull_list, i, (50,150,100))

cv2.imshow('Contours', drawing)

cv2.waitKey(0)
```

результат:



3. Створення обмежувальних рамок і кіл для контурів

КОД:

```
import cv2
import numpy as np

img = cv2.imread("/home/rodion/yuliia0/aboba/cv/cont/cool.jpg")

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

gray = cv2.blur(gray, (3,3))

cv2.imshow('cool', img)

canny= cv2.Canny(gray, 60, 60 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

contours_poly = [None]*len(contours)

boundRect = [None]*len(contours)
```

```
centers = [None]*len(contours)

radius = [None]*len(contours)

for i, c in enumerate(contours):

    contours_poly[i] = cv2.approxPolyDP(c, 3, True)

    boundRect[i] = cv2.boundingRect(cv2.approxPolyDP(c, 3, True))

    centers[i], radius[i] = cv2.minEnclosingCircle(contours_poly[i])



drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

for i in range(len(contours)):

    cv2.drawContours(drawing, contours_poly, i, (150,10,140))

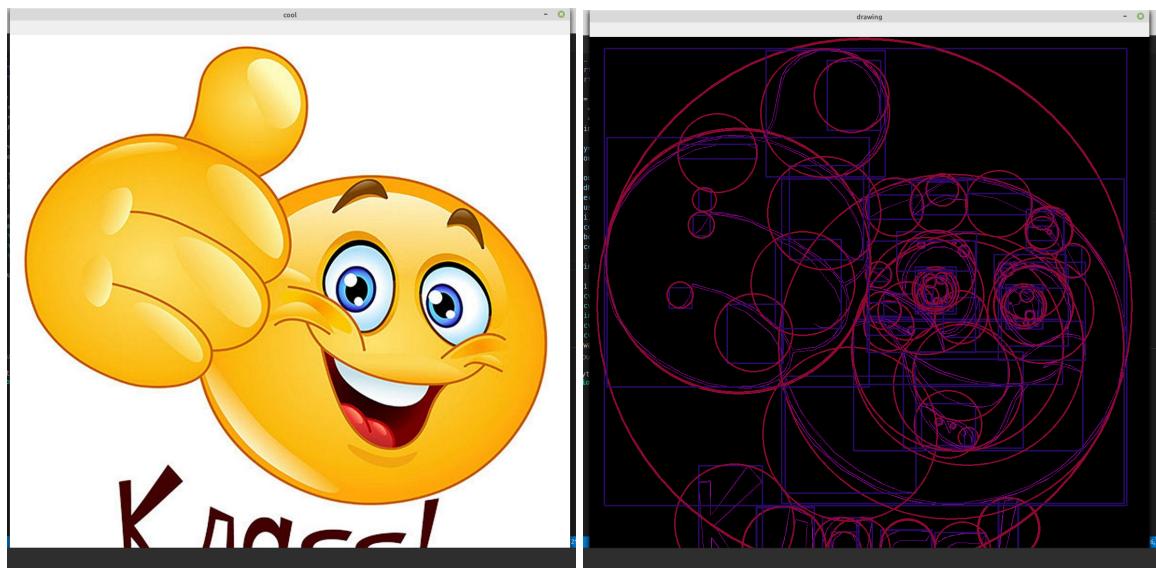
    cv2.rectangle(drawing, (int(boundRect[i][0]), int(boundRect[i][1])),
    (int(boundRect[i][0]+boundRect[i][2]),
    int(boundRect[i][1]+boundRect[i][3])), (100,10,40), 2)

    cv2.circle(drawing, (int(centers[i][0]), int(centers[i][1])),
    int(radius[i]), (50,10,140), 2)

cv2.imshow('drawing', drawing)

cv2.waitKey()
```

результат:



4. Створення обмежувальних поворотних прямокутників і еліпсів для контурів

КОД:

```
import cv2
import numpy as np

img = cv2.imread("/home/rodion/yuliia0/aboba/cv/cont/cool.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = cv2.blur(gray, (3,3))

cv2.imshow('cool', img)

canny= cv2.Canny(gray, 65, 65 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

minRect = [None]*len(contours)

minEllipse = [None]*len(contours)

for i, c in enumerate(contours):
```

```

minRect[i] = cv2.minAreaRect(c)

if c.shape[0] > 5:

minEllipse[i] = cv2.fitEllipse(c)


drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

for i, c in enumerate(contours):

cv2.drawContours(drawing, contours, i, (150,10,140))

cv2.ellipse(drawing, minEllipse[i], (50,100,40), 2)

box = cv2.boxPoints(minRect[i])

box = np.intp(box)

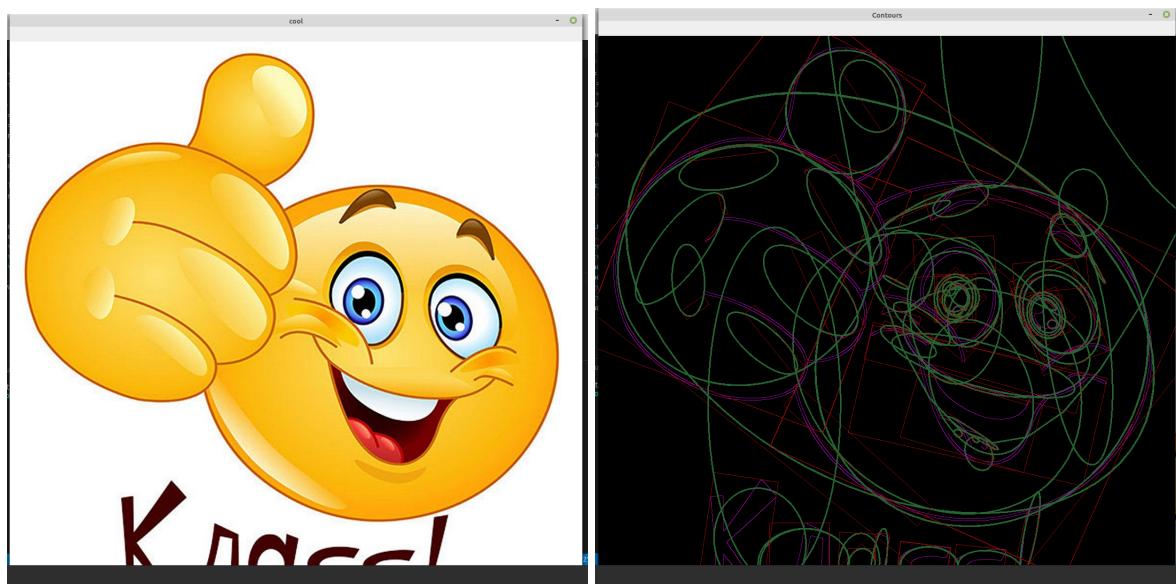
cv2.drawContours(drawing, [box], 0, (10,10,140))

cv2.imshow('Contours', drawing)

cv2.waitKey(0)

```

результат:



5. Моменти зображення

КОД:

```
import cv2

import numpy as np


img = cv2.imread("/home/rodion/yuliia0/aboba/cv/cont/cool.jpg")

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

gray = cv2.blur(gray, (3,3))

cv2.imshow('cool', img)


canny= cv2.Canny(gray, 65, 65 * 2)

contours, hierarchy = cv2.findContours(canny, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)


mc = [None]*len(contours)

mu = [None]*len(contours)

for i in range(len(contours)):

    mu[i] = cv2.moments(contours[i])

    mc[i] = (mu[i]['m10'] / (mu[i]['m00'] + 1e-5), mu[i]['m01'] /
(mu[i]['m00'] + 1e-5))



drawing = np.zeros((canny.shape[0], canny.shape[1], 3), dtype=np.uint8)

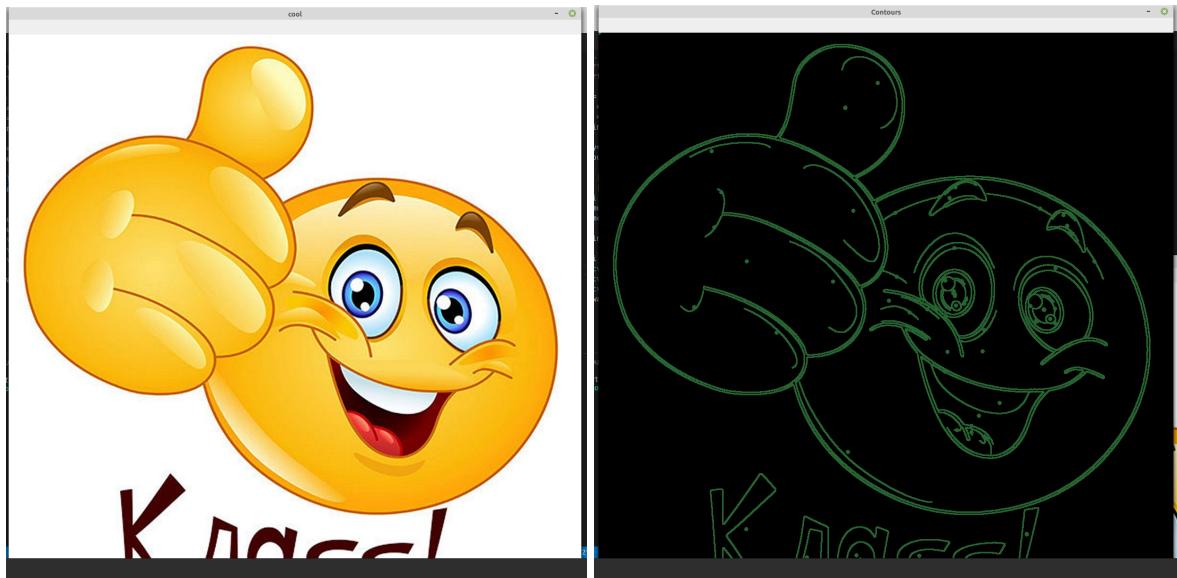
for i in range(len(contours)):

    cv2.drawContours(drawing, contours, i, (50,100,40), 2)

    cv2.circle(drawing, (int(mc[i][0]), int(mc[i][1])), 4, (50,100,40), -1)
```

```
cv2.imshow('Contours', drawing)  
cv2.waitKey(0)
```

результат:



6. Тест «Точковий багатокутник»

код:

результат: