4.Контури

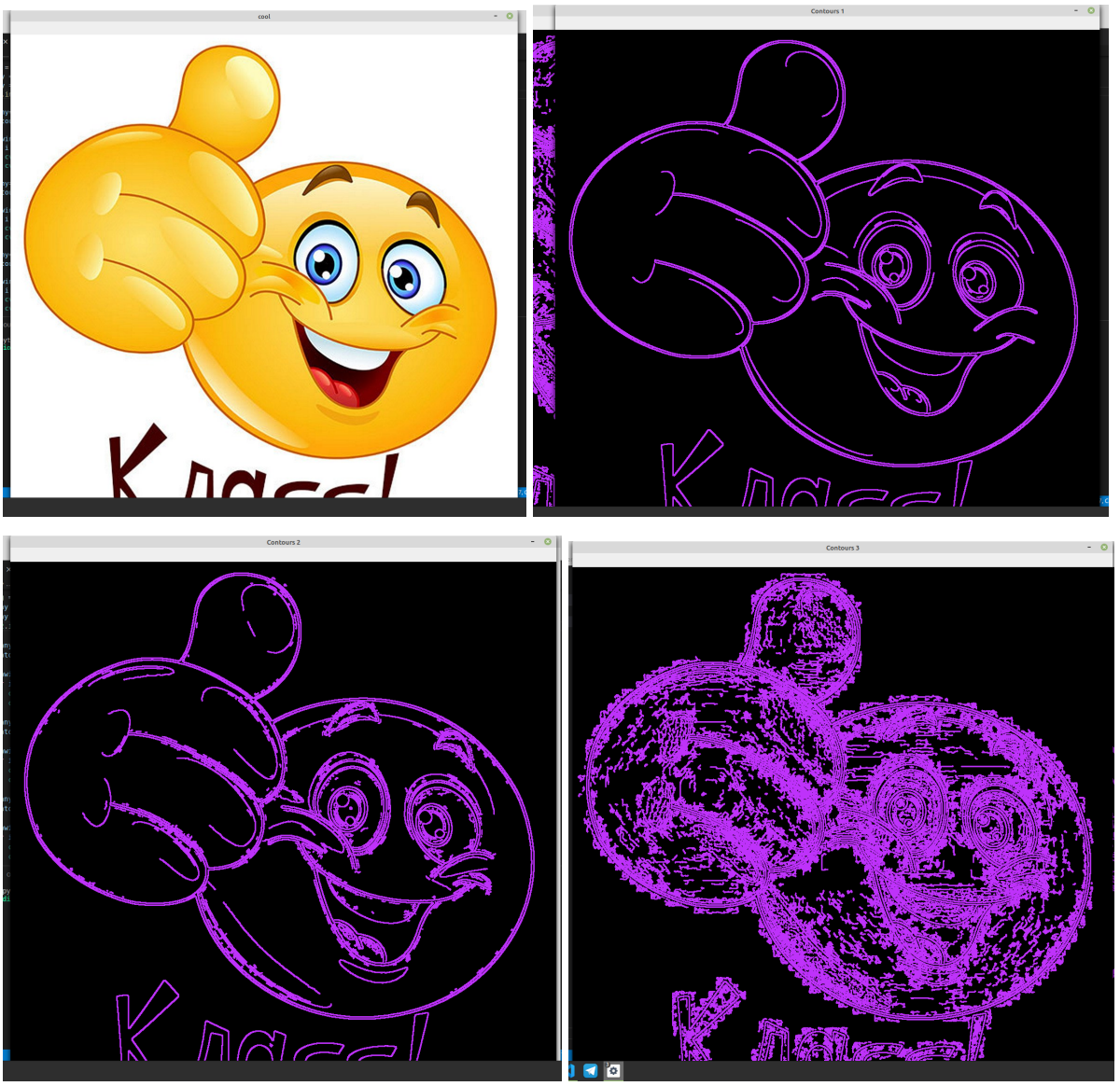
1. Знаходження контурів на зображенні

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:



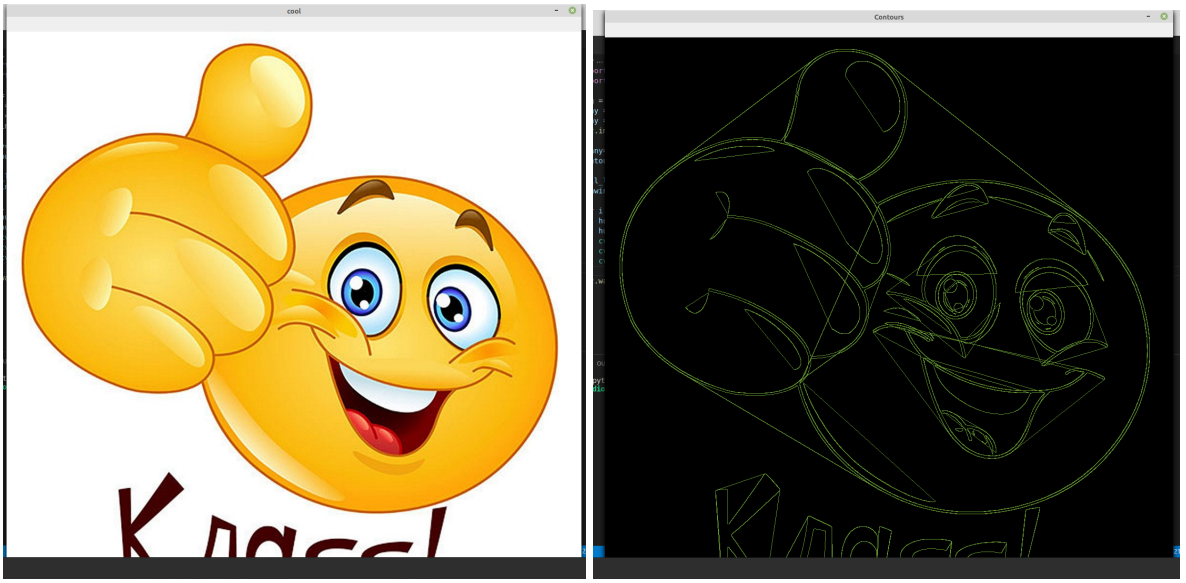
2. Опукла оболонка

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:



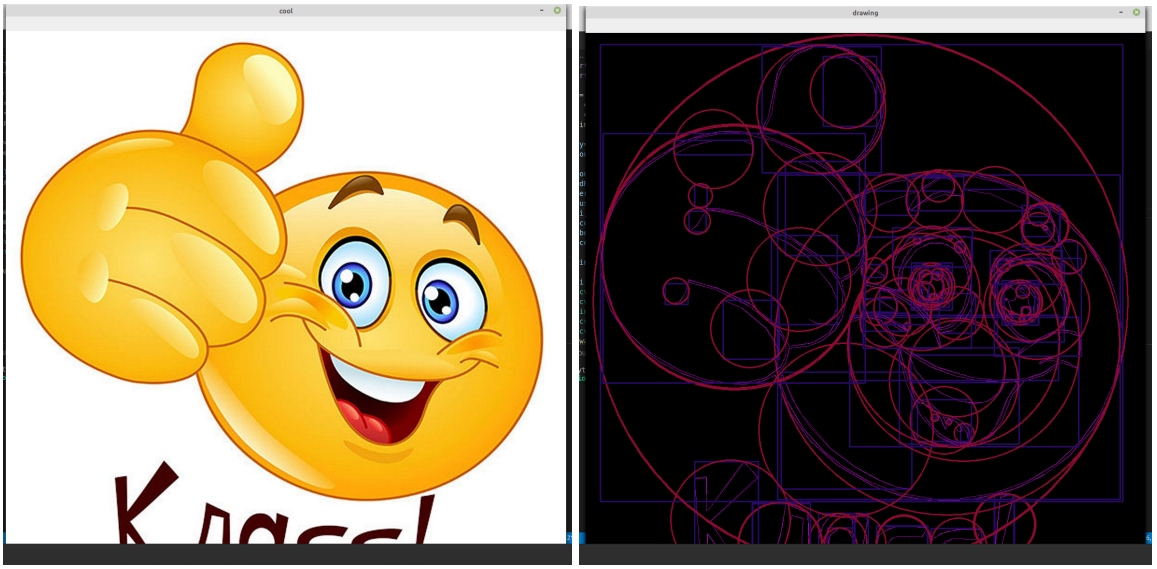
3. Створення обмежувальних рамок і кіл для контурів

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:



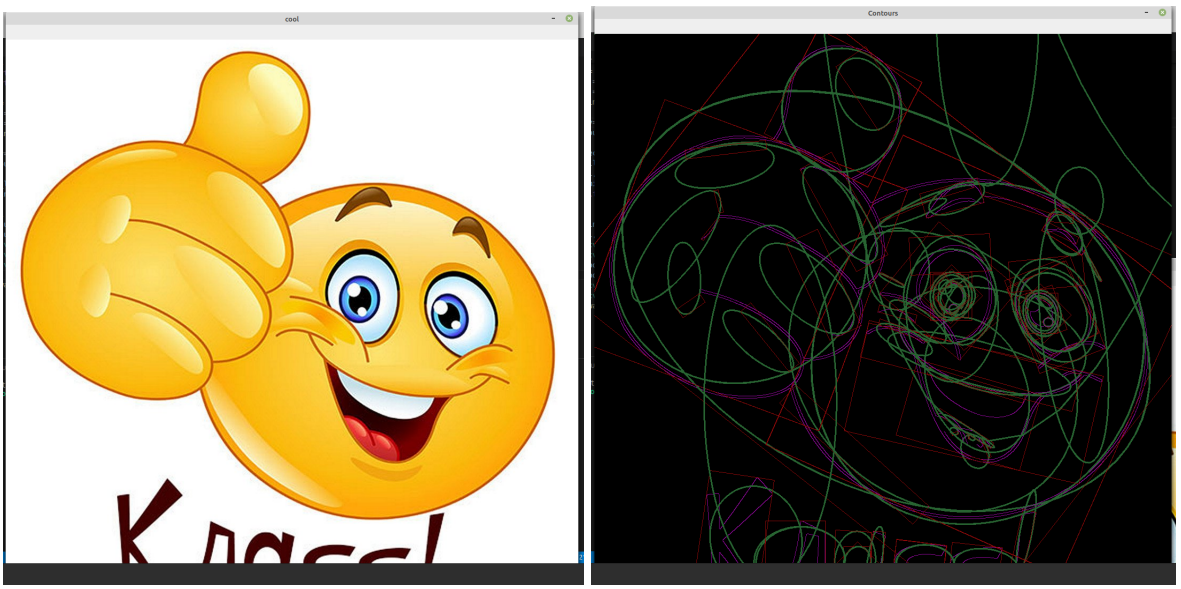
4. Створення обмежувальних поворотних прямокутників і еліпсів для контурів

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:



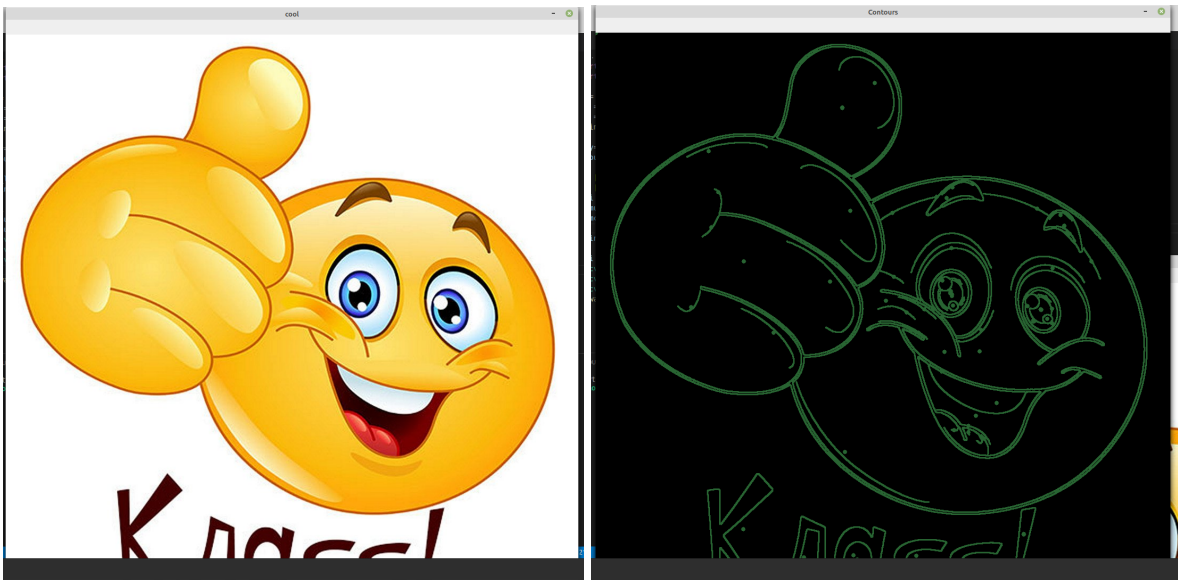
5. Моменти зображення

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»

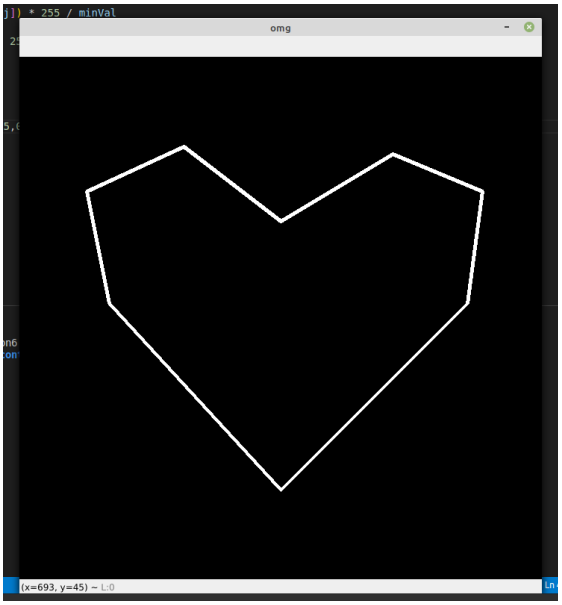
код:

```
import cv2

import numpy as np
```

```
img = np.zeros((700, 700), dtype=np.uint8)


vert = [None]*8

vert[0] = (350, 220)

vert[1] = (220, 120)

vert[2] = (90, 180)

vert[3] = (120, 330)

vert[4] = (350, 580)

vert[5] = (600, 330)

vert[6] = (620, 180)

vert[7] = (500, 130)


for i in range(8):

cv2.line(img, vert[i], vert[(i+1)%8], ( 255 ), 3)


contours, _ = cv2.findContours(img, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)


raw = np.empty(img.shape, dtype=np.float32)

for i in range(img.shape[0]):

for j in range(img.shape[1]):

raw[i,j] = cv2.pointPolygonTest(contours[0], (j,i), True)


minVal, maxVal, _, maxDistPt = cv2.minMaxLoc(raw)

minVal = abs(minVal)
```

```
maxVal = abs(maxVal)



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

for i in range(img.shape[0]):

for j in range(img.shape[1]):

if raw[i,j] < 0:

drawing[i,j,0] = 255 - abs(raw[i,j]) * 255 / minVal

elif raw[i,j] > 0:

drawing[i,j,2] = 255 - raw[i,j] * 255 / maxVal

else:

drawing[i,j,0] = 255

drawing[i,j,1] = 255

drawing[i,j,2] = 255



cv2.circle(drawing,maxDistPt, int(maxVal),(255,0,255), 1, cv2.LINE_8,
0)



cv2.imshow('omg', img)

cv2.imshow('cool drawing', drawing)

cv2.waitKey(0)
```

результат:


j]) * 255 / minVal
omg
(x=693, y=45) ~ L:0


cool drawing
(x=685, y=74) ~ R:0 G:0 B:162