

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

Звіт
з лабораторної роботи №4
з дисципліни «Комп'ютерний зір»
«Виділення габаритних розмірів об'єкта»

Виконав ст. гр. ІПЗм-22-7:
Миронюк С.А.

Перевірів викладач:
Работягов А.В.

Харків, 2023

Виділення габаритних розмірів об'єкта

Мета завдання:

- якомога точніше (по можливості) виділити габаритні розміри та визначити центр об'єкта;
- для об'єкта відобразити вікно "захоплення" об'єкта та центр об'єкта.

Виділення розміру об'єкта реалізується шляхом знаходження контурів навколо нього. Це можна зробити, спочатку завантаживши зображення об'єкта, перетворивши його на відтінки сірого та застосувавши поріг для відокремлення об'єкта від фону. Потім він знаходить контури об'єкта у пороговому зображенні та малює їх на вихідному зображенні. Код обчислює площу об'єкта в пікселях за допомогою функції `cv2.contourArea()` і перетворює площу в реальні одиниці виміру з використанням коефіцієнта масштабування. Нарешті, він друкує розмір об'єкта у вибраній одиниці виміру.

OpenCV — це бібліотека з відкритим вихідним кодом, яка спрямована на комп'ютерне бачення в реальному часі. Ця бібліотека розроблена компанією Intel і є кросплатформною – вона може підтримувати Python, C++, Java тощо. Комп'ютерне бачення – це найсучасніша галузь комп'ютерних наук, яка має на меті дозволити комп'ютерам розуміти, що видно на зображенні. OpenCV є однією з найбільш широко використовуваних бібліотек для завдань комп'ютерного бачення, таких як розпізнавання обличчя, виявлення руху, виявлення об'єктів тощо.

Хід роботи

Вхідне зображення:



Код для виділення габаритних розмірів об'єкта:

```
import cv2 # Імпортується бібліотека (OpenCV) для роботи із зображеннями
import numpy # Імпортується бібліотека для чисельних розрахунків, аналізу та візуалізації даних на основі мови програмування Python (вона може створювати різні матриці)

img = cv2.imread('images/Blood Cell.jpg') # Створимо змінну яка звертається до бібліотеки та її методу читання зображення де параметром є зображення

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Змінна звертається до бібліотеки OpenCV та її методу конвертування кольори де аргументами є картинка і метод перевагає зображення
# із формату BGR у чорно білий

cv2.imshow('image', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
cv2.waitKey(0) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)

ret, thresh = cv2.threshold(
    gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU) # Застосовується threshold попір до зображення, щоб відокремити об'єкти від фону

contours, hierarchy = cv2.findContours(
    thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) # Метод findContours до якого звертаємось через бібліотеку OpenCV повертає значення у дві змінні (у першу
# встановлюється список з усіма позиціями контурів, а в другу встановлюється ієрархія самих об'єктів)

for cnt in contours: # Цикл що, переглядає контури та обчислює площу кожного об'єкта
    area = cv2.contourArea(cnt)

    x, y, w, h = cv2.boundingRect(cnt) # cv2.boundingRect() - це функція, яка використовується для створення приблизно прямокутника разом із зображенням.
    # Основне призначення цієї функції - виділити цікаву область після отримання зовнішньої форми зображення

    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2) # Через бібліотеку звертаємось до методу, що малює квадрат із заданими параметрами (наша змінна, (початок
    # квадрата по ширині, початок квадрата за висотою), (кінець квадрата за шириною, кінець квадрата за висотою), (копір квадрата BGR))

cv2.imshow('image', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
cv2.waitKey(2000) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)

print(f"Габаритні розміри об'єкта: ширина дорівнює {w} висота дорівнює {h}") # Консольне виведення для обчислення габаритних розмірів об'єкта (ширини та об'єкта)

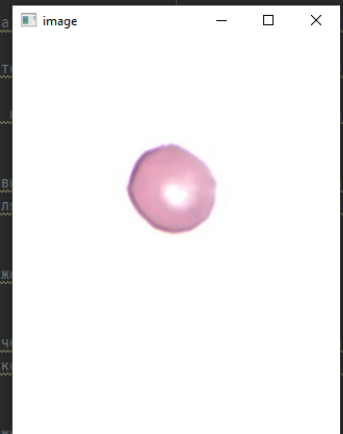
cv2.circle(img, (x + (w//2), y + (h//2)), 3, (255, 0, 0), thickness=cv2.FILLED) # Через бібліотеку звертаємось до методу, що малює коло із заданими параметрами

cv2.imshow('image02', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
cv2.waitKey(2000) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)

print(f"Центр виділеного об'єкта знаходиться у координатах: x = {x + (w//2)} y = {y + (h//2)}") # Консольне виведення координат де знаходиться центр об'єкта
```

Програма спочатку показує вхідне зображення:

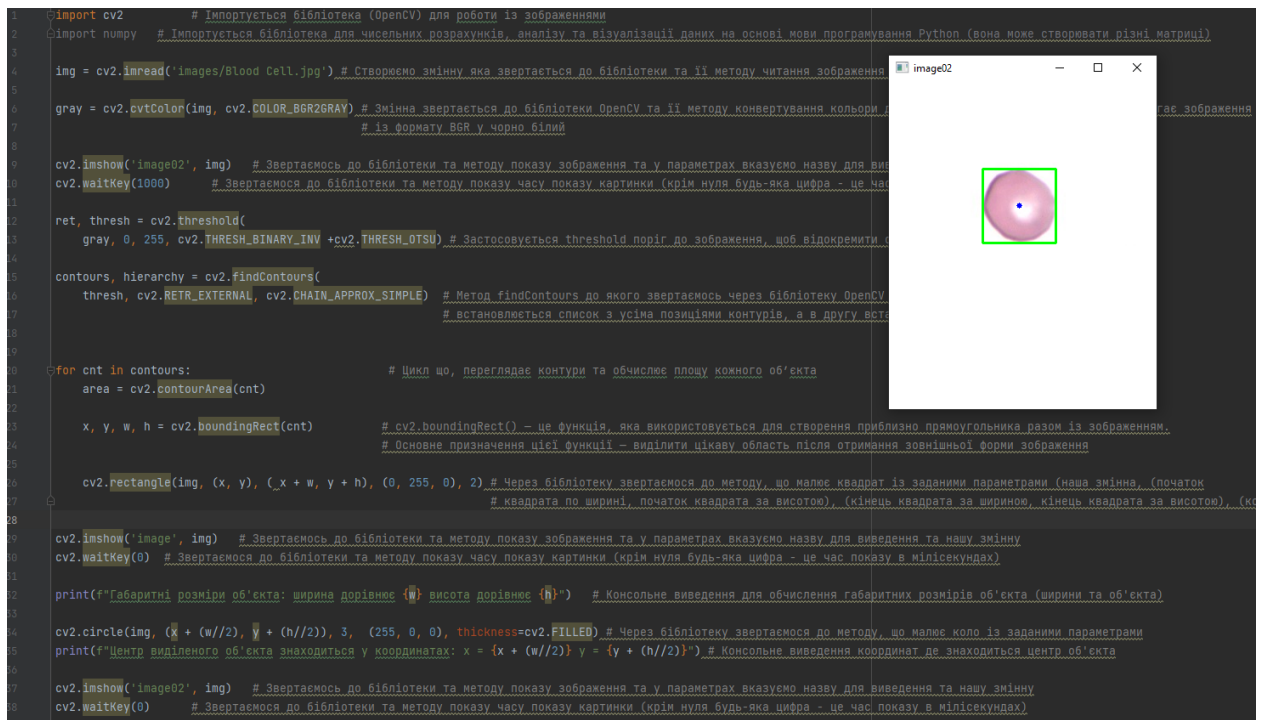
```
1 import cv2 # Імпортується бібліотека (OpenCV) для роботи із зображеннями
2 import numpy # Імпортується бібліотека для чисельних розрахунків, аналізу та візуалізації даних на основі мови програмування Python (вона може створювати різні матриці)
3
4 img = cv2.imread('images/Blood Cell.jpg') # Створимо змінну яка звертається до бібліотеки та її методу читання зображення де параметром є зображення
5
6 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Змінна звертається до бібліотеки OpenCV та її методу конвертування кольори де аргументами є картинка і метод перевагає зображення
7 # із формату BGR у чорно білий
8
9 cv2.imshow('image', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
10 cv2.waitKey(0) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)
11
12 ret, thresh = cv2.threshold(
13     gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU) # Застосовується threshold попір до зображення, щоб відокремити об'єкти від фону
14
15 contours, hierarchy = cv2.findContours(
16     thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) # Метод findContours до якого звертаємось через бібліотеку OpenCV повертає значення у дві змінні (у першу
17 # встановлюється список з усіма позиціями контурів, а в другу встановлюється ієрархія самих об'єктів)
18
19 for cnt in contours: # Цикл що, переглядає контури та обчислює площу кожного об'єкта
20     area = cv2.contourArea(cnt)
21
22     x, y, w, h = cv2.boundingRect(cnt) # cv2.boundingRect() - це функція, яка використовується для створення приблизно прямокутника разом із зображенням.
23     # Основне призначення цієї функції - виділити цікаву область після отримання зовнішньої форми зображення
24
25     cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2) # Через бібліотеку звертаємось до методу, що малює квадрат із заданими параметрами (наша змінна, (початок
26     # квадрата по ширині, початок квадрата за висотою), (кінець квадрата за шириною, кінець квадрата за висотою), (копір квадрата BGR))
27
28 cv2.imshow('image', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
29 cv2.waitKey(2000) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)
30
31 print(f"Габаритні розміри об'єкта: ширина дорівнює {w} висота дорівнює {h}") # Консольне виведення для обчислення габаритних розмірів об'єкта (ширини та об'єкта)
32
33 cv2.circle(img, (x + (w//2), y + (h//2)), 3, (255, 0, 0), thickness=cv2.FILLED) # Через бібліотеку звертаємось до методу, що малює коло із заданими параметрами
34
35 cv2.imshow('image02', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
36 cv2.waitKey(2000) # Звертаємось до бібліотеки та методу показу часу показу картини (крім нуля будь-яка цифра - це час показу в мілісекундах)
37
38 print(f"Центр виділеного об'єкта знаходиться у координатах: x = {x + (w//2)} y = {y + (h//2)}") # Консольне виведення координат де знаходиться центр об'єкта
39
```



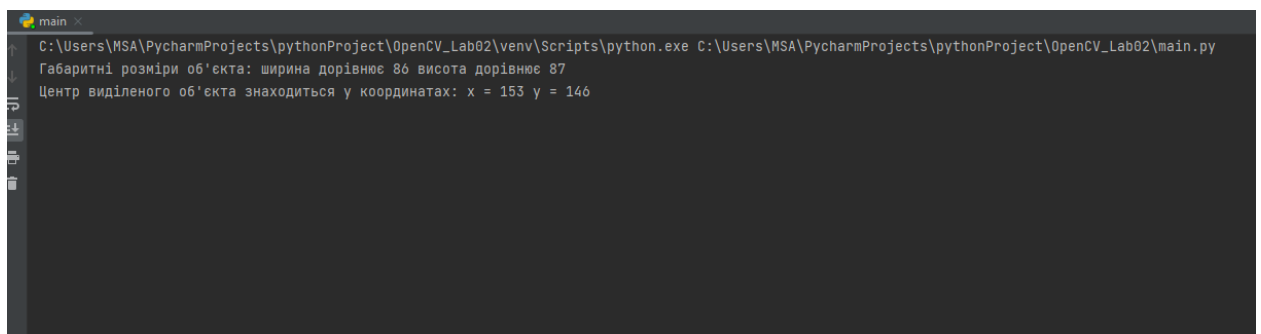
Програма показує обведення об'єкта квадратом:



Програма показує обведення об'єкта квадратом і центр об'єкта:



В консолі відображається: чому дорівнює висота і ширина об'єкта, а також обчислення центру виділеного об'єкта у координатах:



Висновок: в результаті виконаної роботи навчився виділяти габаритні розміри об'єкта, відображати центр об'єкта та розраховувати координати об'єкта.