

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

Звіт
з лабораторної роботи №2
з дисципліни «Комп'ютерний зір»
на тему «Пошук об'єктів заданої категорії»

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

Перевірив викладач:
Работягов А.В.

Харків, 2023

Пошук об'єктів заданої категорії

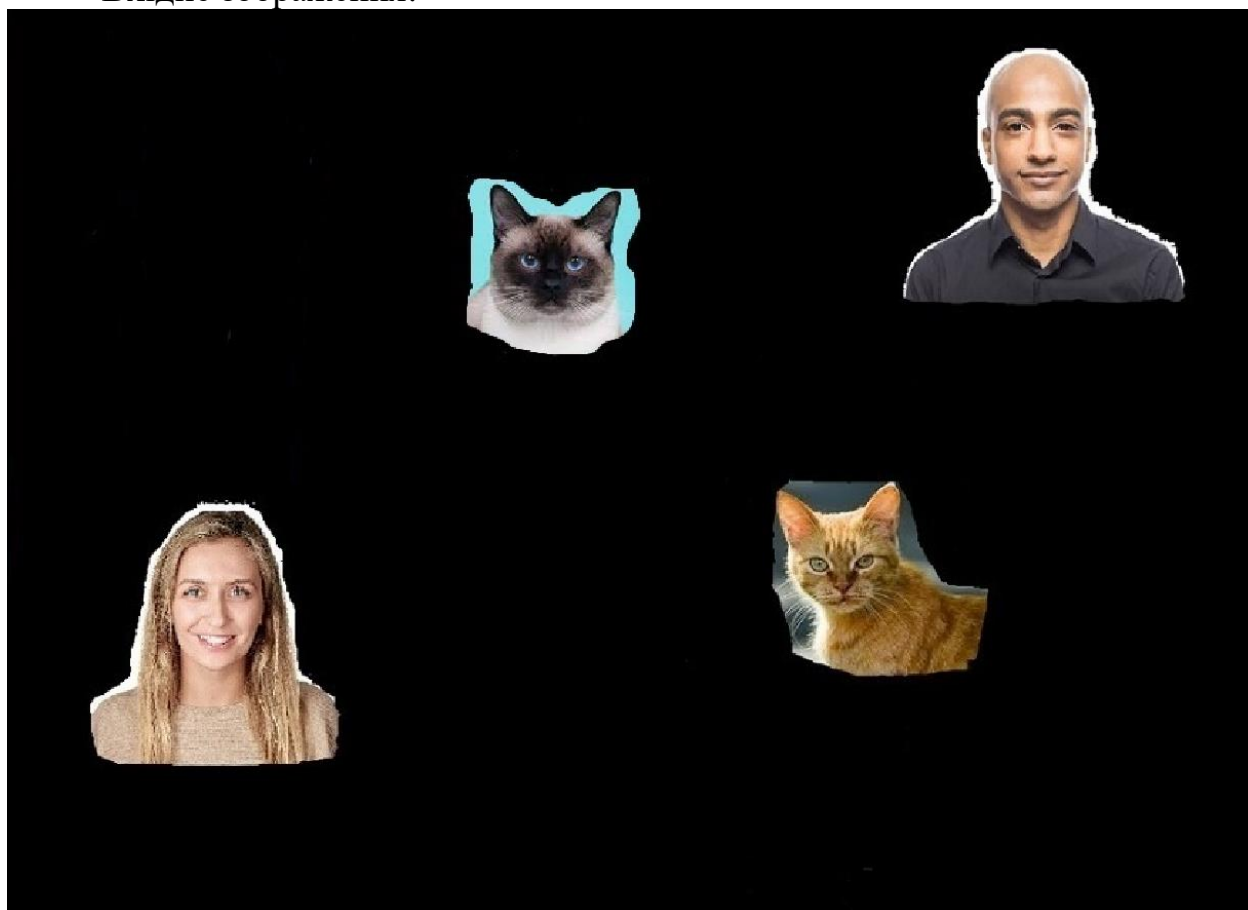
Мета завдання:

- ознайомитися з пристроєм систем пошуку об'єктів певної категорії на фотографіях;
- реалізувати простий варіант такої системи;
- отримати досвід інтегрування сторонніх бібліотек.

Пошук об'єктів заданої категорії – одне з ключових завдань комп'ютерного зору, що має важливе практичне застосування. Швидкі та відносно надійні алгоритми пошуку осіб зараз вбудовані у більшість фотоапаратів. Активно розвиваються системи виявлення пішоходів та автомобілів.

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.

Вхідне зображення:

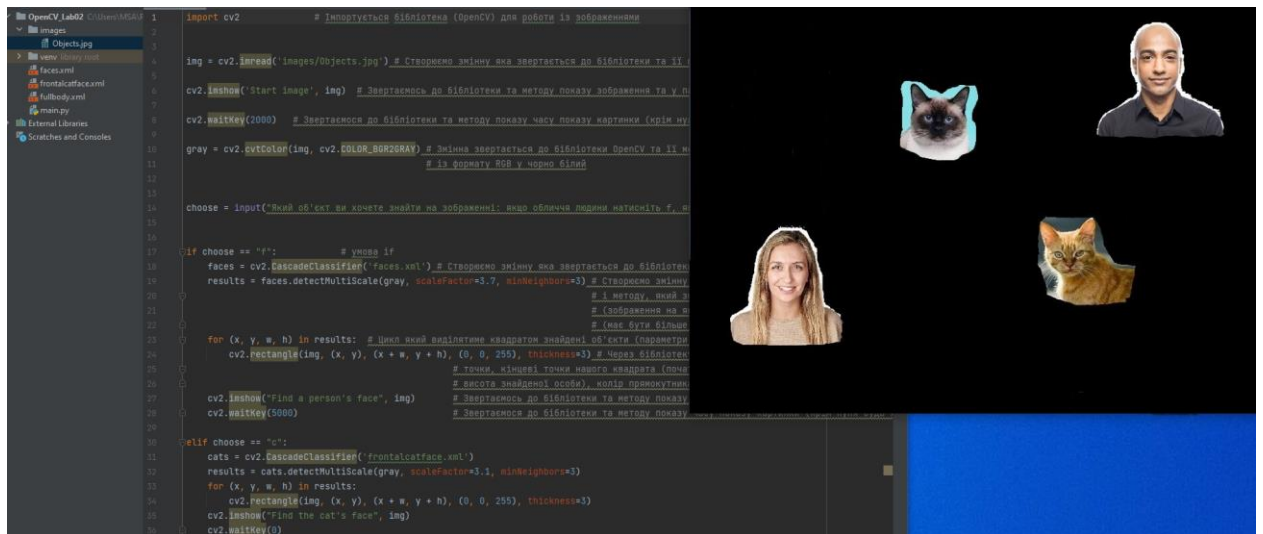


Код для виявлення конкретного об'єкту:

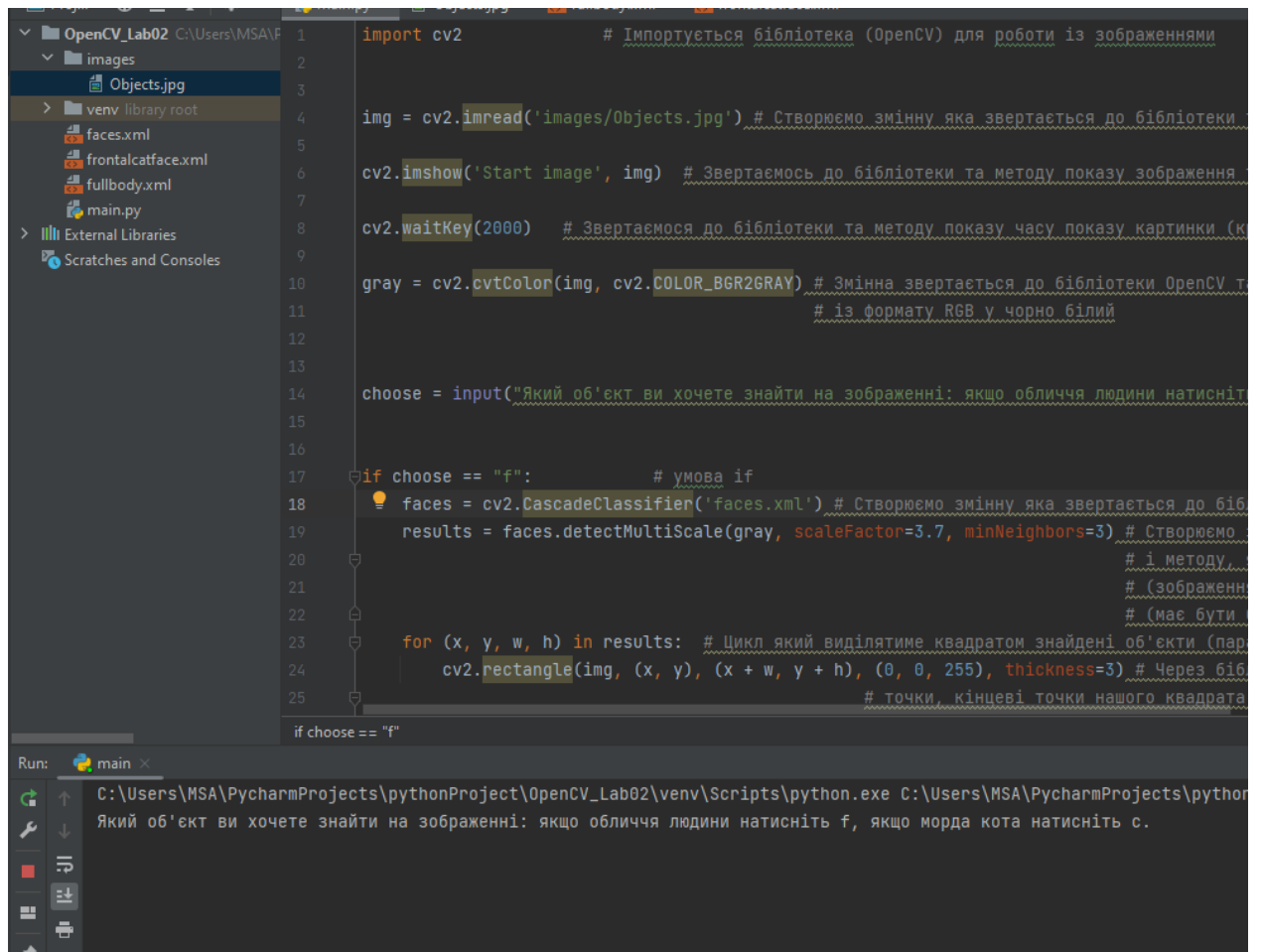
```
Project - OpenCV_Lab02 C:\Users\MSA\PycharmProjects\
  images
  Objects.jpg
  faces.xml
  frontalcatface.xml
  fullbody.xml
  main.py
  External Libraries
  Scratches and Consoles

1 import cv2 # Імпортується бібліотека (OpenCV) для роботи із зображеннями
2
3
4 img = cv2.imread('images/Objects.jpg') # Створимо змінну яка звертається до бібліотеки та її методу читання зображення де параметром є зображення
5
6 cv2.imshow('Start image', img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
7
8 cv2.waitKey(2000) # Звертаємось до бібліотеки та методу показу часу показу картинки (крім нуля будь-яка цифра - це час показу в мілісекундах)
9
10 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Змінна звертається до бібліотеки OpenCV та її методу конвертування кольорів де вругметами є картинка і метод передає дає зображення
11 # із формату RGB у чорно білий
12
13
14 choose = input("Який об'єкт ви хочете знайти на зображенні: якщо обличчя людини натисніть f, якщо морда кота натисніть c\n") # Консольне введення
15
16
17 if choose == "f": # умова if
18     faces = cv2.CascadeClassifier('faces.xml') # Створимо змінну яка звертається до бібліотеки та її методу який бере якийсь файл і витягує його як натреновану модель параметром файл xml
19     results = faces.detectMultiScale(gray, scaleFactor=3.7, minNeighbors=3) # Створимо змінну яка буде зберігати координати знайдених осіб. Змінна звертається до натренованої моделі
20 # і методу, який знаходить координати всіх знайдених об'єктів (тобто осіб) при цьому задаються параметри
21 # (зображення на якому будемо все шукати, параметр адаптує модель під розміри об'єктів на зображенні
22 # (має бути більше 1), параметр визначальний скільки подібних об'єктів може бути один біля одного
23
24 for (x, y, w, h) in results: # Цикл який виводитиме квадратом знайдені об'єкти (параметри циклу є позиція за координатами x, позиція за координатами y, ширина, висота) у нашому списку
25     cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), thickness=3) # Через бібліотеку звертаємось до методу малювання прямокутників з параметрами (наше зображення, початкові
26 # точки, кінцеві точки нашого квадрата (початкова координата по X плюс ширина знайденої особи, початкова координата Y плюс
27 # висота знайденої особи), колір прямокутника у форматі BGR (беремо червоний [0]), ширина обведення
28 cv2.imshow("Find a person's face", img) # Звертаємось до бібліотеки та методу показу зображення та у параметрах вказуємо назву для виведення та нашу змінну
29 cv2.waitKey(5000) # Звертаємось до бібліотеки та методу показу часу показу картинки (крім нуля будь-яка цифра - це час показу в мілісекундах)
30
31 elif choose == "c":
32     cats = cv2.CascadeClassifier('frontalcats.xml')
33     results = cats.detectMultiScale(gray, scaleFactor=3.1, minNeighbors=3)
34     for (x, y, w, h) in results:
35         cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), thickness=3)
36     cv2.imshow("Find the cat's face", img)
37     cv2.waitKey(0)
```

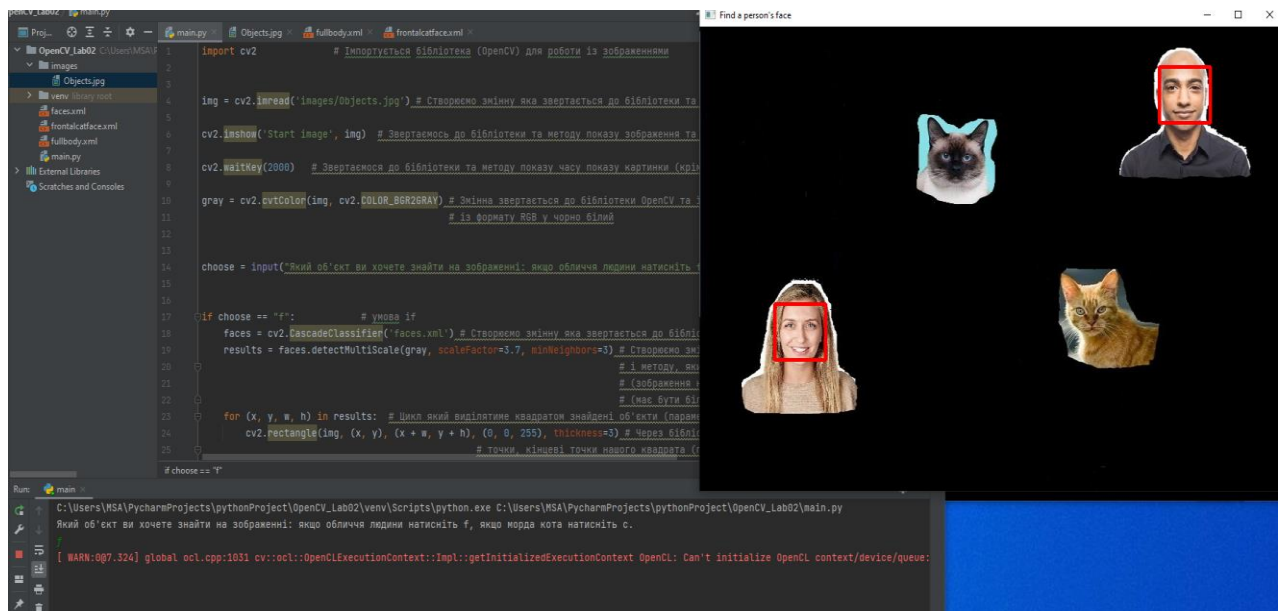
Програма спочатку показує вхідне зображення:



Далі у консолі показує питання користувачу:

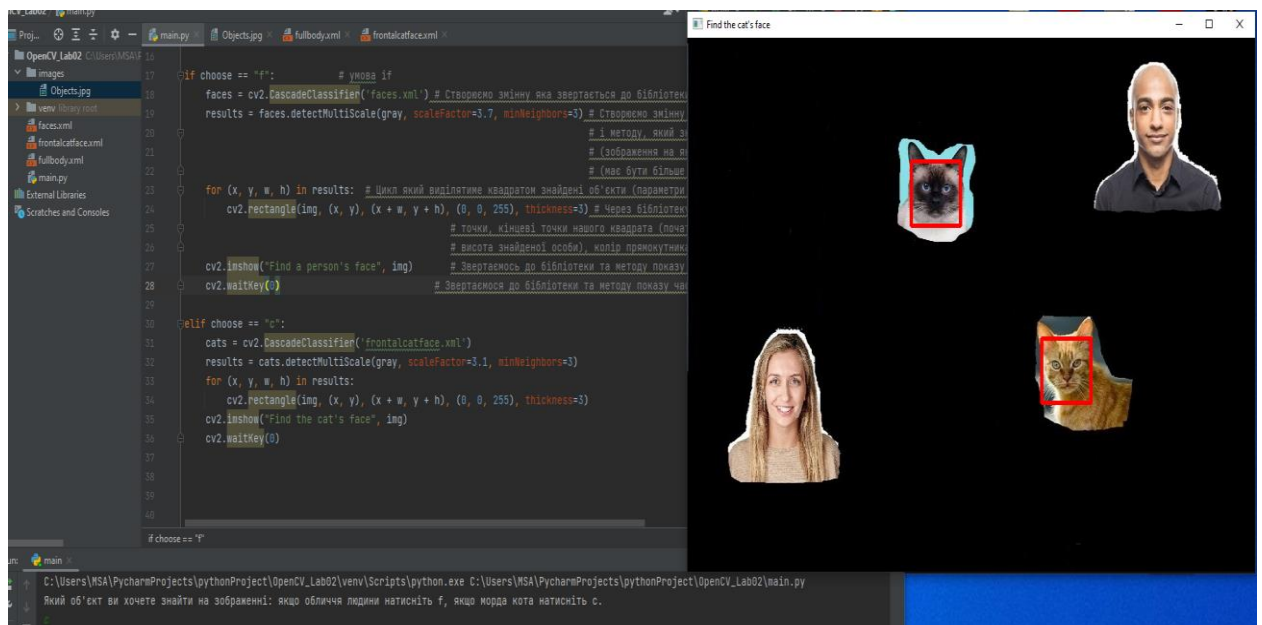


Користувач обирає щоб програма знайшла обличчя людини, робота коду:



Користувач обирає щоб програма знайшла морду кота.

Робота коду:



Висновок: розробив код для пошуку об'єктів певної категорії на зображенні, прописав коментарі до коду.