

Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки



Кафедра інформаційних систем та технологій

Модульна контрольна робота з дисципліни «Технології Computer Vision»

Виконала:
студентка групи ІС-12
Павлова Софія

Перевірив:
Писарчук О. О.

Київ – 2024

1. Завдання

Варіант №22:

Білет №10

1. Фрактальні зображення.
2. Дескриптор зображення.
3. Здійснити розробку програмного скрипта з виявлення об'єктів за алгоритмами каскадів Хаара.

2. Виконання

2.1. Теоретичне питання № 1

Фрактальні зображення

Фрактальні зображення базуються на імітації оригінальних рішень живої та неживої природи – повторення і тиражування вдалих комбінацій в структурі об'єкту.

Фрактальне зображення може бути векторним або піксельним, до яких застосована модель їх структури, яка об'єднана в єдине і неподільне ціле як **елементарний фрактал**. Далі фрактальне зображення повторюється і тиражується по аналогії з головними парадігмами ООП.

Отже, *для створення фрактального зображення використовують фрактали*.
Уведемо це поняття.

Фракталом, за *Мандельбротом*, називається геометрична фігура, яка складається з частин, які в певному розумінні подібні за формою на всю фігуру.

Тобто це геометрична фігура, в якій один і той самий фрагмент повторюється при кожному зменшенні масштабу (частина об'єкта схожа на сам об'єкт).

Математично фрактал можна представити наступним чином:

$((\text{растр} + \text{вектор}) * \text{кофіцієнт трансформації}) * \text{кофіцієнт повторення}$

Приклад *природного фракталу*:



Рисунок 1 – Природний фрактал

Приклад *фракталу*:

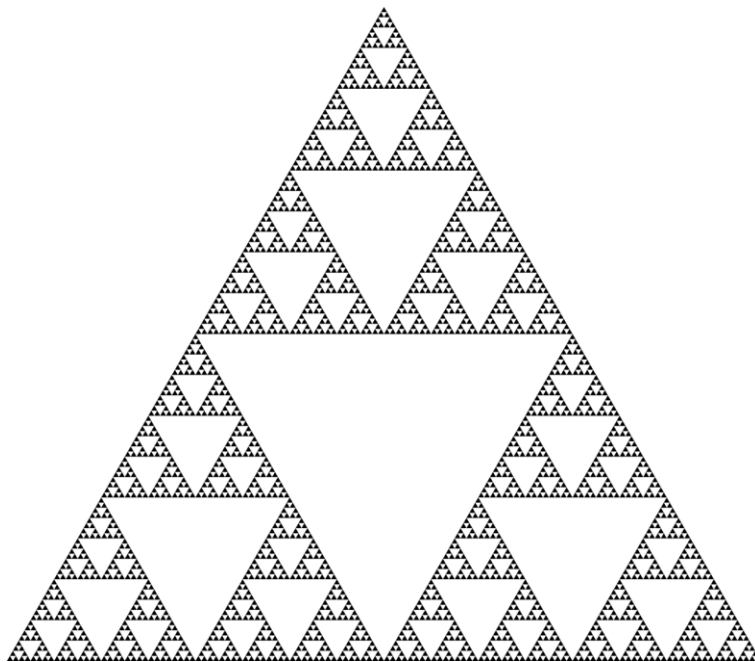


Рисунок 2 – Приклад фрактала

Приклад *фрактального зображення*:



Рисунок 3 – Приклад фрактального зображення

2.2. Теоретичне питання № 2

Дескриптор зображення

Для початку уведемо поняття особливої точки.

Почнімо з того, що для виділення із зображення інформації, що може бути **унікальною для ідентифікації**, необхідно прив'язатися до локальних особливостей зображення – **особливих точок**.

Особлива точка t зображення – це точка зображення, околицю якої $o(t)$ можна відрізнити від околиці будь-якої іншої точки зображення $o(n)$ в деякій іншій околиці особливої точки $o_2(t)$.

Тоді **дескриптор** – це ідентифікатор особливої точки, що виділяє її з решти.

Одними з прикладів дескриптора зображень є *дескриптор кутової точки Харріса*, або *SIFT* – *масштабно-інваріативна трансформація ознак*.

2.3. Практичне завдання

Завдання

Розробка програмного скрипта з виявлення об'єктів за алгоритмами каскадів Хаара.

Постановка задачі

Використовуючи будь-яку фотографію з декількома людьми, виявити на ньому обличчя, очі, усмішку. Порахувати кількість осіб на фото.

Програмна реалізація

Для цього використаємо каскади Хаара для обличчя, очей та усмішки.

Алгоритм програми:

За алгоритмом Віюлі-Джонса виявимо фронтально розташовані обличчя на фото. Далі для кожного обличчя знайдемо очі та усмішки, використовуючи відповідні каскади пакету *OpenCV*.

Підрахуємо кількість осіб на фото як кількість розпізнаних облич.

Лістинг коду:

```
import cv2
image = cv2.imread('photo.jpg')

# Зменшення розміру зображення до 50% від оригінального розміру
scale_percent = 50 # відсотків оригінального розміру
width = int(image.shape[1] * scale_percent / 100)
height = int(image.shape[0] * scale_percent / 100)
image = cv2.resize(image, (width, height), interpolation=cv2.INTER_AREA)

# Ініціалізація класифікаторів для виявлення облич, очей та усмішок
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_eye.xml')
smile_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_smile.xml')
# Конвертація зображення у відтінки сірого
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Виявлення облич на зображенні
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=2, minSize=(85,
85))
num_faces = len(faces)
# Виявлення очей та усмішок
for (x, y, w, h) in faces:
```



```

cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
roi_gray = gray[y:y + h, x:x + w]
roi_color = image[y:y + h, x:x + w]
# Виявлення очей
eyes = eye_cascade.detectMultiScale(roi_gray, scaleFactor=1.001, minNeighbors=30,
minSize=(1, 1))
for (ex, ey, ew, eh) in eyes:
    cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)
# Виявлення усмішки
smiles = smile_cascade.detectMultiScale(roi_gray, scaleFactor=1.01, minNeighbors=10,
minSize=(30, 30))
for (sx, sy, sw, sh) in smiles:
    cv2.rectangle(roi_color, (sx, sy), (sx + sw, sy + sh), (0, 0, 255), 2)

print("Кількість осіб на фото:", num_faces)
cv2.imshow('Face Detection', image)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Результат:

При правильно підібраних параметрах для зображення, алгоритм розпізнає всі обличчя, очі та більшість усмішок.

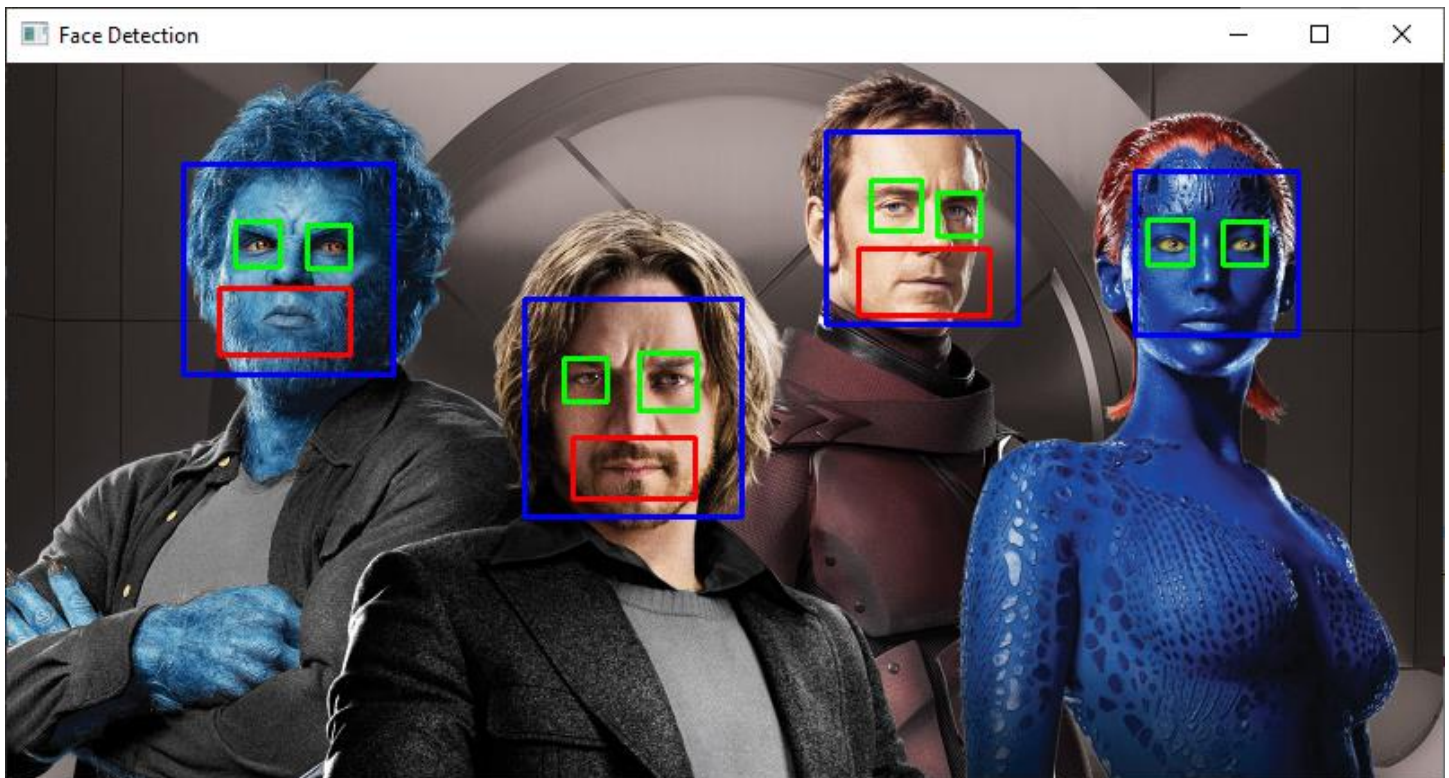
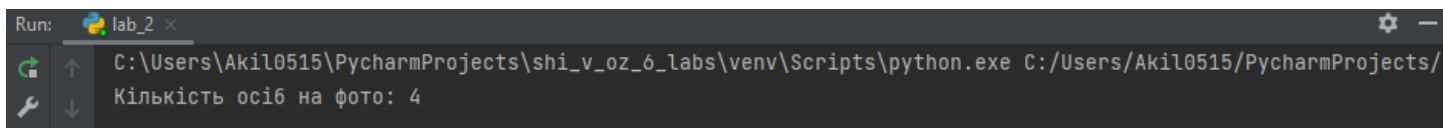


Рисунок 4 – Розпізнавання облич, очей та усмішок на фото

Бачимо також, що алгоритм правильно порахував кількість осіб на фото.



```
Run: lab_2 x
C:\Users\Akil0515\PycharmProjects\shi_v_oz_6_labs\venv\Scripts\python.exe C:/Users/Akil0515/PycharmProjects/
Кількість осіб на фото: 4
```

Рисунок 2 – Підрахунок кількості осіб на фото

Висновок:

Під час виконання модульної контрольної роботи було сформульовано прикладну задачу з власного досвіду для цифрової обробки зображень.

Розроблено програмний скрипт для виявлення облич, очей та усмішок на цифровому зображенні. Реалізовано функцію підрахунку кількості осіб на фото. Ідентифікація облич, очей та усмішок реалізована з використанням технології каскадів Хаара.

Програма успішно апробована на тестовому зображенні.