

Міністерство освіти і науки України

Національний технічний університет України „КПІ імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Звіт до лабораторної роботи №2

З дисципліни «Штучний інтелект в задачах обробки зображень»

|  |  |
| --- | --- |
| Перевірив:  доц. Нікітін В.А. | Виконала:  Студентка 3 курсу, гр. ІС-13  Росновська О.О. |

Київ 2024 р.

**Лабораторна робота №2.**

**Тема:** Виявлення об’єктів засобами OpenCV

**Мета:** навчитись виявляти обличчя та пішоходів в режимі реального часу за допомогою OpenCV

**Завдання:**

1. Використовуючи будь-яку фотографію з декількома людьми, виявити на ньому обличчя, очi, усмiшку. Порахувати кiлькiсть осiб на фото;
2. Зробити розпізнавання використовуючи будь-яке відео з обличчям людини, тривалiстю не менше 30 секунд. Можно використати камеру ноутбука;
3. Обробити вiдеофал, так щоб вiн видiляв пiшоходiв i, по можливостi, їхнi обличчя. Файл можна взяти з youtube i вирiзати ролик тривалiстю не менше 30 секунд.

# Завдання 1

Використовуючи будь-яку фотографію з декількома людьми, виявити на ньому обличчя, очi, усмiшку. Порахувати кiлькiсть осiб на фото;

Для цього завдання я скористалась алгоритмом Віоли-Джонса. Це алгоритм для виявлення обличчя та інших об'єктів на зображеннях. Він шукає на зображенні певні піксельні маски, які відповідають певним особливостям об'єкта, що необхідно знайти.

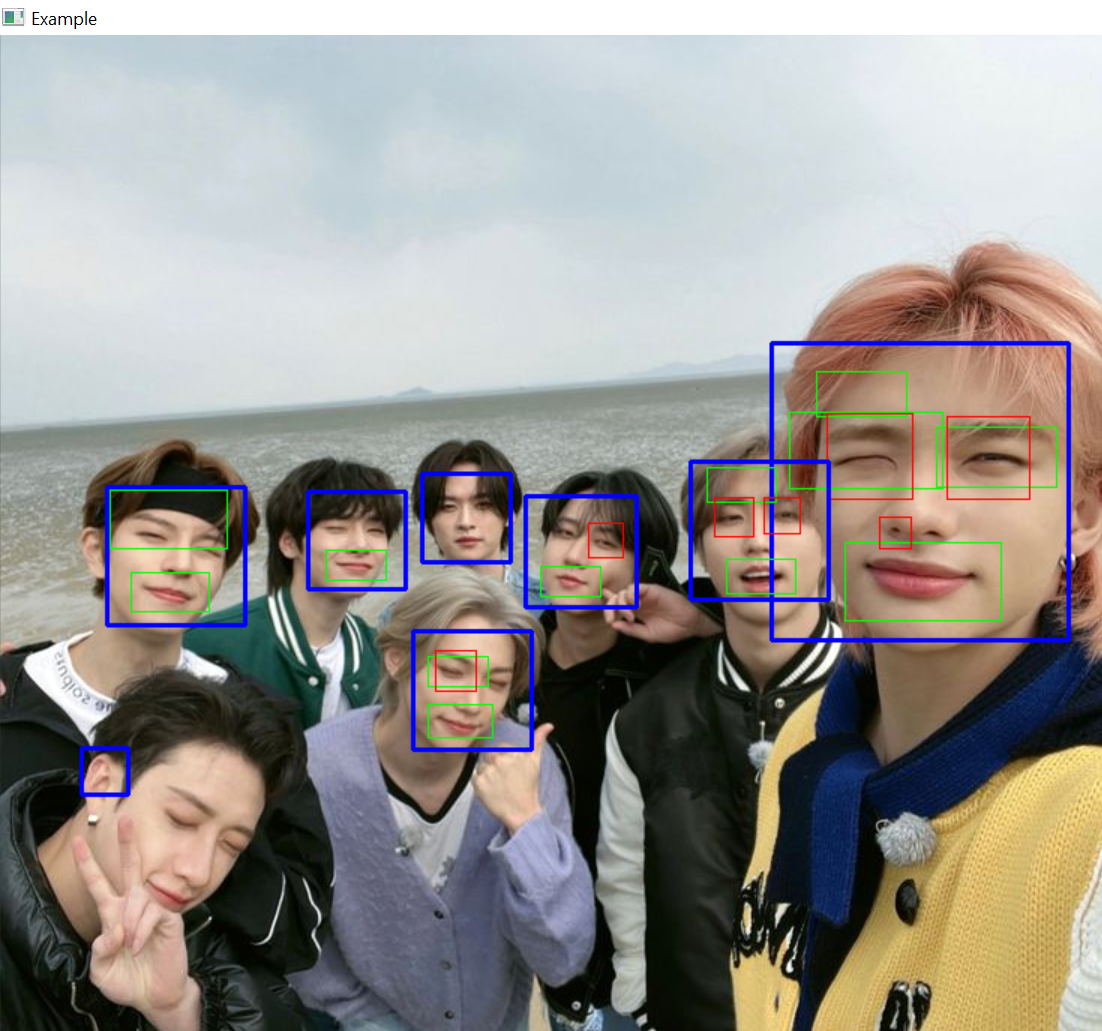
На жаль даний алгоритм не здатен виявити обличчя що не розташовано в анфас. Це і є його основний недолік.

Необхідно завантажити класифікатори, за якими якраз алгоритм і виявляє об’єкти, а саме класифікатори обличчя, посмішки та очей. Далі необхідно зчитати вихідне зображення та перетворити його в ЧБ. Тепер можна застосувати класифікатор облич:

face\_rects=face\_cascade.detectMultiScale(gray\_filter, scaleFactor=1.1, minNeighbors=5)

Розпізнати обличчя це звісно добре, але головне відобразити їх графічно. Для цього циклом проходимось по кожному знайденому обличчю та малюємо прямокутник з його координатами.

Аби оптимізувати процес пошуку очей та усмішки, зробимо це в циклі. Адже можна створити виділену зону меншого розміру і шукати в ній. Аналогічно проходячись циклом по знайденим об’єктам малюємо графічно прямокутники.



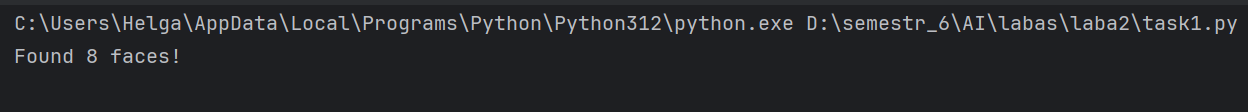


Рис.1 Результат програми Завдання 1

Отже, програма знайшла 8/8 обличь, проте зі знаходженням одного з них просто повезло. Вухо зарахувало як обличчя.

Посмішки програма віднайшла досить добре, а от очі ні. Програма рідко бачить щоб схоже на око, навіть майже не хибила.



Рис.2 Детальне зображення

Хоча тут ми маємо ну майже ідеальний анфас – очі та посмішку не розпізнано, скоріш за все через чубчик.

# Завдання 2

Зробити розпізнавання використовуючи будь-яке відео з обличчям людини, тривалiстю не менше 30 секунд. Можно використати камеру ноутбука;

Майже той самий код, єдина різниця в тому, що замість зображення ми маємо послідовність зображень – відео. Тут ми проходимось по кожному фрейму в циклі while.

cap = cv2.VideoCapture('video.mp4')

while True:  
 ret, frame = cap.read()

|  |  |
| --- | --- |
|  |  |
|  |  |

Рис.3 Результат програми

# Завдання 3

Обробити вiдеофал, так щоб вiн видiляв пiшоходiв i, по можливостi, їхнi обличчя. Файл можна взяти з youtube i вирiзати ролик тривалiстю не менше 30 секунд.

Для виявлення пішоходів застосуємо дескриптор HOG:

hog = cv2.HOGDescriptor()  
hog.setSVMDetector(cv2.HOGDescriptor\_getDefaultPeopleDetector())

Пошук обличчя відбувається за аналогічно, тільки відбувається він всередині знайдених пішоходів (жовтих прямокутників). Це допомагає запобігти хибним знахідкам.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рис.4 Результат програми