

**Міністерство освіти і науки України  
Національний технічний університет України «КПІ» імені Ігоря Сікорського  
Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ  
із залікової роботи  
з навчальної дисципліни «Вступ до технології Data Science»**

**Виконав:**

Студент 3 курсу кафедри ОТ ФІОТ,  
Навчальної групи ІМ-13  
Тавлуй Д. О.

**Перевірив:**

Професор кафедри ОТ ФІОТ  
Писарчук О.О.

**Київ 2024**

## **I. Білет №21**

### **II. Завдання:**

1. Що впливає на точність оцінювання експериментальної вибірки?
2. Яка етапність алгоритмів прийняття рішень?
3. Розробити програмний скрипт, що здійснює розпізнавання людини.

### **III. Результати виконання залікової роботи.**

#### **3.1. Відповідь на теоретичне питання №1.**

Основна вимога до експериментальної вибірки - репрезентативність щодо досліджуваної популяції. Усі особливості популяції, з якою пов'язана перевірювана робоча гіпотеза, повинні мати своє адекватне представництво в вибірці. Також, розмір вибірки має велике значення. Зазвичай більші вибірки дають більш точні оцінки. Зменшення розміру вибірки може призвести до великої варіабельності результатів і менш точних оцінок параметрів.

В деяких випадках випадкові події можуть впливати на результати дослідження. Наприклад, якщо в експериментальній групі відбувається подія, яка впливає на результат, то це може вплинути на точність оцінки.

Якщо вибірка не є випадковою, це може викликати систематичні помилки в оцінюванні. Важливо, щоб кожен елемент популяції мав однаковий шанс потрапити в вибірку.

Якісність і точність вимірювань також може впливати на оцінювання вибірки. Невірно вимірювані або неточні дані можуть призвести до неточностей в аналізі.

Використання правильних статистичних методів для аналізу даних також важливе.

Неправильне використання методів може призвести до недостовірних результатів.

Узагальнення результатів на всю популяцію вимагає обережності, оскільки деякі вибіркові висновки можуть бути неадекватними. Загалом, мета полягає в максимізації точності та вірогідності отримання достовірних результатів за умови врахування цих факторів.

#### **3.2. Відповідь на теоретичне питання №2.**

Алгоритм прийняття рішень - це систематичний підхід до вирішення проблеми або вибору оптимальної стратегії. Існують різні моделі алгоритмів прийняття рішень, але загальний набір етапів включає в себе наступні:

1. Визначення основної мети або цілі, якої досягнення є основним завданням прийняття рішення.
2. Оцінка зовнішніх і внутрішніх чинників, які можуть впливати на вибір рішення.
3. Розробка чітких критеріїв, які визначають, як буде оцінюватися ефективність різних альтернатив.
4. Розгляд і вибір можливих варіантів чи стратегій для вирішення проблеми.
5. Визначення остаточного вибору, який відповідає встановленим критеріям та досягає цілі ефективності.

Ці етапи можуть варіюватися в залежності від конкретної моделі прийняття рішень і можуть бути виконані в різному порядку або повторюватися в процесі розв'язання проблем.

### 3.3. Відповідь на практичне питання №3.

#### 3.3.1. Математична модель.

Для розпізнавання людини, а саме мого обличчя, я використав бібліотеку cv2, а також класифікатор «haarcascade\_frontalface\_default.xml» з цієї ж бібліотеки.

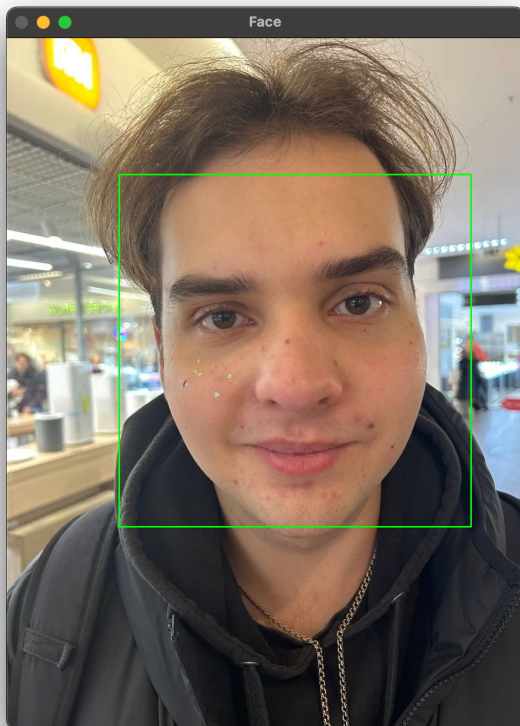
Спочатку фотографія завантажується, потім відбувається конвертація у відтінки сірого. Створюється об'єкт класифікатора каскаду для виявлення обличчя, для цього використовується згаданий раніше xml файл. Застосовується класифікатор каскаду для виявлення обличчя на фотографії сірого відтінку. І вже потім програма перебирає знайдені обличчя визначає координати прямокутника, який охоплює обличчя.

#### 3.3.2. Опис архітектурних рішень та структури проекту.

Я використав монолітну архітектуру проектування, бо сама програма не дуже складна та громізка. Уся логіка реалізована у одному файлі під назвою main.py.

#### 3.3.3. Результати роботи програми відповідно до завдання.

Отримав такий результат:



Як можна побачити, моє обличчя на фотографії розпізналось добре. Вважаю, що результат чудовий.

### 3.3.4. Програмний код, що забезпечує отримання результату.

```
import cv2

image = cv2.imread('me.jpg')

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')

faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3,
minNeighbors=5)

for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow('Face', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**Виконав студент: Тавлуй Дмитро Олександрович, група ІМ-13**