

ПРИДНІПРОВСЬКА ДЕРЖАВНА АКАДЕМІЯ БУДІВНИЦТВА ТА АРХІТЕКТУРИ

Кафедра комп'ютерних наук, інформаційних технологій
та прикладної математики

Лабораторна робота № 4

з дисципліни «Нейронні мережі»

Виконав(ла) студент(ка) III курсу

група КНЗ-22 спеціальність 122
«Комп'ютерні науки»

_____Пойманова Дар'я_____
(прізвище та ініціали)

Переві́рив

(прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

м. Дніпро – 2025

Зміст

ВСТУП	3
ЛАБОРАТОРНА РОБОТА №4	4
РОЗПІЗНАВАННЯ ОБЛИЧЧЯ.....	4
1. Розпізнавання обличчя	4
2. Розпізнавання обличчя з відеопотоку	5
3. Розпізнавання обличчя з записаного відео	7
КОНТРОЛЬНІ ЗАПИТАННЯ.....	8
ВИСНОВОК	11

Вступ

Розпізнавання обличчя є однією з найбільш затребуваних технологій у сучасному світі, яка використовується у багатьох сферах, включаючи безпеку, біометричну аутентифікацію, маркетинг та інтерактивні застосунки. З розвитком комп'ютерного зору та нейронних мереж цей процес став набагато ефективнішим та точнішим. Python завдяки великій кількості бібліотек, зокрема OpenCV, надає широкі можливості для реалізації алгоритмів розпізнавання облич.

Метою даної лабораторної роботи є ознайомлення з методами розпізнавання обличчя, використовуючи бібліотеку OpenCV мовою програмування Python. В рамках роботи було розглянуто та реалізовано процеси розпізнавання облич з фотографій, потоків веб-камери та відеофайлів. Основними завданнями є налаштування програмного середовища, створення програмних кодів та аналіз ефективності отриманих алгоритмів.

Під час виконання лабораторної роботи були використані методи на основі каскадів Хаара, які дозволяють швидко та якісно виявляти обличчя на зображеннях різної складності. Це дозволило детально ознайомитися з особливостями роботи цих алгоритмів у різних умовах освітлення, кутах зйомки та інших факторах, що впливають на точність розпізнавання.

ЛАБОРАТОРНА РОБОТА №4

РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

Мета роботи: навчитися створювати програми для розпізнавання обличчя мовою Python за допомогою бібліотеки OpenCV.

1. Розпізнавання обличчя

```
import cv2
```

```
# Завантаження класифікатора Haarcascade
face_cascade_db = cv2.CascadeClassifier(cv2.data.haarcascades +
"haarcascade_frontalface_default.xml")

# Завантаження зображення
img = cv2.imread("photo2.jpg")

# Перетворення зображення у відтінки сірого
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Розпізнавання обличчя
faces = face_cascade_db.detectMultiScale(img_gray, scaleFactor=1.1,
minNeighbors=5)

# Відображення рамок на зображенні
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

# Виведення результату
cv2.imshow('Detected Faces', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Результат роботи програми:



Рисунок 1. Результат розпізнавання обличчя на фотографії

2. Розпізнавання обличчя з відеопотоку

Необхідно написати програму, яка здатна розпізнавати обличчя з потоку веб-камери.

```
import cv2

face_cascade_db = cv2.CascadeClassifier(cv2.data.haarcascades +
"haarcascade_frontalface_default.xml")

# Захоплення потоку з веб-камери
cap = cv2.VideoCapture(0)

# Перевірка чи успішно відкрилась веб-камера
if not cap.isOpened():
    print("Помилка: не вдалося підключитися до камери.")
    exit()

while True:
    # Отримання кадру
    success, img = cap.read()

    # Перевірка отримання кадру
    if not success:
        print("Помилка: не вдалося отримати кадр із камери.")
        break

    # Конвертація у сірий відтінок
```

```

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Розпізнавання обличч
faces = face_cascade_db.detectMultiScale(img_gray, scaleFactor=1.1,
minNeighbors=5)

# Виведення рамок навколо обличч
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

# Виведення результату
cv2.imshow('Webcam Face Detection', img)

# Вихід з циклу при натисканні 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Звільнення ресурсів
cap.release()
cv2.destroyAllWindows()

```

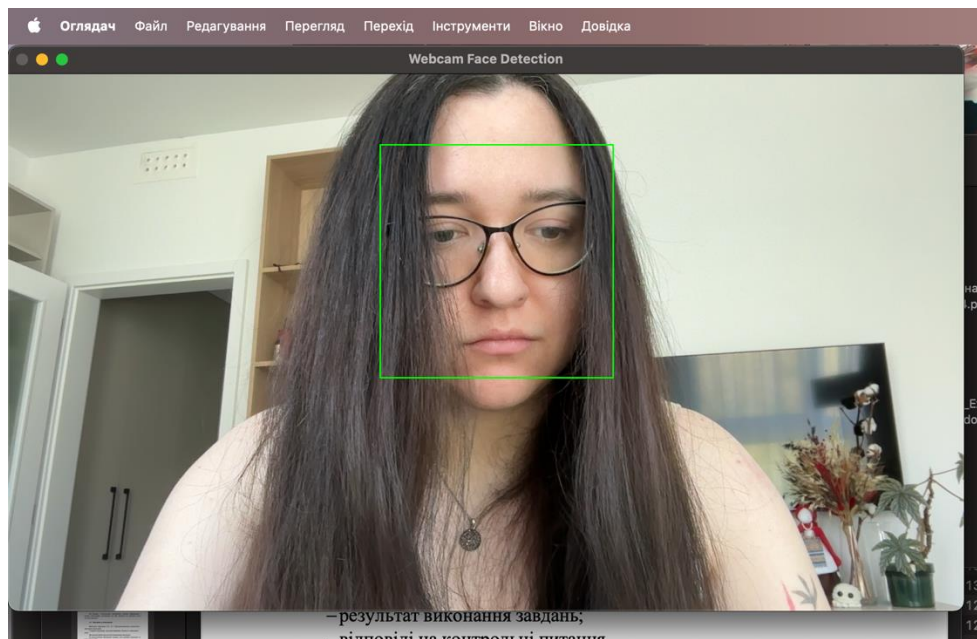


Рисунок 2. Результат розпізнавання обличчя з потоку веб-камери

Як бачимо з результатів, програмою зчитано інформацію з потоку веб-камери, розпізнано на ній обличчя та обведено його зеленою рамкою.

3. Розпізнавання обличчя з записаного відео

Створити програму для розпізнавання обличчя із записаного відео.

```
import cv2

face_cascade_db = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_frontalface_default.xml")

# Завантаження відео-файлу
cap = cv2.VideoCapture("video2.mp4")

while cap.isOpened():
    success, frame = cap.read()
    if not success:
        break

    img_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade_db.detectMultiScale(img_gray, scaleFactor=1.1,
minNeighbors=5)

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

    cv2.imshow('Face Detection from Video', frame)

    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Результат роботи програми:



Рисунок 3. Розпізнавання обличчя з відео 1

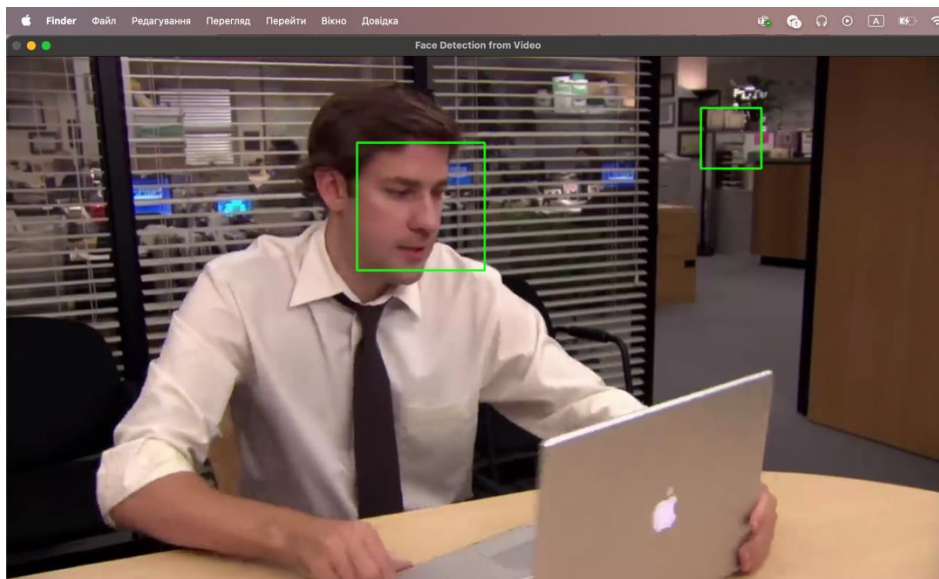


Рисунок 4. Розпізнавання обличчя з відео 2

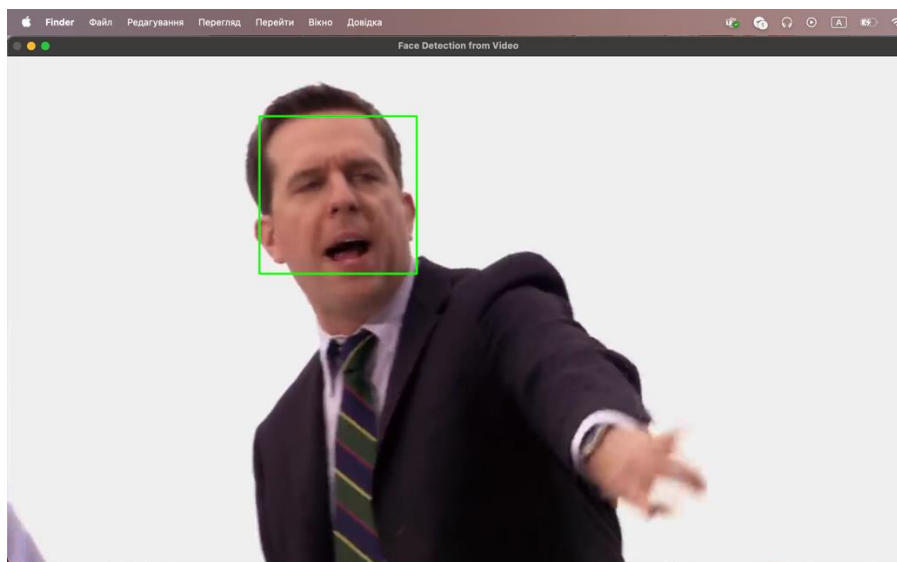


Рисунок 5. Розпізнавання обличчя з відео 3

Контрольні запитання

1. Програмні продукти для розпізнавання облич

FaceID (Apple) – біометрична система для авторизації користувача на пристроях Apple.

DeepFace (Facebook) – алгоритм для автоматичного тегування фотографій.

Face++ – хмарна платформа для ідентифікації обличчя, аналізу та біометрії.

Amazon Rekognition – сервіс Amazon для аналізу зображень, в тому числі розпізнавання облич.

Microsoft Azure Face API – хмарний сервіс для ідентифікації облич, визначення віку, статі та емоцій.

OpenFace – відкрите програмне рішення для розпізнавання облич із відкритим кодом.

2. Бібліотеки Python для розпізнавання

OpenCV – найпоширеніша бібліотека комп'ютерного зору, підтримує методи Хаара для детектування облич.

Dlib – ефективна бібліотека, що має інструменти для розпізнавання та порівняння облич.

face_recognition – проста бібліотека на базі dlib, що дозволяє швидко розпізнавати та ідентифікувати обличчя.

MTCNN – бібліотека для точного виявлення та визначення ключових точок на обличчі.

DeepFace – потужна бібліотека для аналізу та ідентифікації облич на основі глибокого навчання.

3. Бібліотеки Python для роботи з нейронними мережами

TensorFlow – найбільш популярна бібліотека для створення нейронних мереж, що підтримує глибоке навчання.

Keras – високорівневий інтерфейс до TensorFlow, що дозволяє швидко створювати нейронні мережі.

PyTorch – бібліотека з динамічним обчислювальним графом, зручна для розробки та досліджень.

Caffe – швидка бібліотека для згорткових нейронних мереж, переважно використовується в комп'ютерному зорі.

Scikit-learn – базові алгоритми для нейронних мереж, переважно для задач класичного машинного навчання.

4. Особливості згорткових нейронних мереж (CNN)

- CNN спеціалізовані для роботи з візуальними даними (зображення, відео).
- Використовують згорткові шари для виділення характерних ознак об'єктів.
- Зменшують кількість параметрів завдяки загальним вагам (спільні ваги, weight sharing).
- Інваріантні до зсувів і деформацій зображень, завдяки чому мають високу точність.
- Використовують шари пулінгу (pooling layers) для зменшення розмірності та зниження вимог до обчислювальних ресурсів.
- Широко застосовуються для класифікації зображень, розпізнавання облич, аналізу медичних зображень і автономних транспортних засобів.

Висновок

У ході виконання лабораторної роботи було реалізовано і протестовано алгоритми розпізнавання облич за допомогою Python та бібліотеки OpenCV. Експериментально встановлено, що каскади Хаара є ефективним методом для розпізнавання облич на фото, веб-камері та у відеопотоці. Алгоритм демонструє високу швидкість роботи, що робить його придатним для застосування у реальному часі.

Використання різних джерел даних (фото, відео, веб-камера) дозволило виявити певні особливості алгоритму. Наприклад, розпізнавання на фотографіях, зазвичай, є більш точним, ніж у потоках відео, де рух та зміна освітлення можуть суттєво впливати на якість результату. Водночас належне налаштування параметрів алгоритму дозволяє підвищити його ефективність навіть у складних умовах.

Отже, результати лабораторної роботи підтвердили, що Python у поєднанні з OpenCV є потужним інструментом для вирішення задач розпізнавання обличчя. Подальші дослідження можуть включати вдосконалення алгоритмів, використання нейронних мереж та розширення сфер застосування технології розпізнавання.