

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.8:004.72.056.523

ПОГОДЖЕНО

Декан факультету

Інформаційних технологій

_____/ Глазунова О.Г., д.п.н, проф. /

підпис

ПІБ, вчене звання і ступінь

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри

Комп'ютерних систем, мереж та кібербезпеки

_____/ Касаткін Д.Ю., к.п.н., доцент. /

підпис

ПІБ, вчене звання і ступінь

«__» _____ 2022 р.

«__» _____ 2022 р.

МАГІСТЕРСЬКА РОБОТА

На тему: «Розробка комп'ютерної системи розпізнавання обличчя
людей в реальному часі з використанням штучного інтелекту»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: Комп'ютерні системи та мережі

Керівник дипломного проекту: _____ / Волошин С.М. /
підпис ПІБ

Виконав: _____ / Сальник Д.С. /
підпис ПІБ

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

«ЗАТВЕРДЖУЮ»

завідувач кафедри

комп'ютерних систем, мереж та кібербезпеки

/ Касаткін Д.Ю., к.п.н., доцент. /

підпис

ПІБ, вчене звання і ступінь

«__» _____ 2022 р.

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Сальнике Дмитру Сергійовичу

(прізвище, ім'я, по батькові)

Спеціальність (напрямок підготовки): 123 «Комп'ютерна інженерія».

Освітня програма: комп'ютерні системи та мережі

Тема магістерської роботи: «Розробка комп'ютерної системи розпізнавання обличчя людей в реальному часі з використанням штучного інтелекту»

затверджена наказом ректора НУБІП України від “1” листопада 2023 № 1999 "С"

Термін подання завершеної роботи на кафедру _____

Вихідні дані до магістерської роботи: бібліотека комп'ютерного зору з відкритим кодом OpenCV, бібліотека для машинного навчання TensorFlow, Інтерфейс програмування додатків API, FaceRecognition, Dlib

Перелік питань, що підлягають дослідженню:

1. Аналіз предметної області для дослідження методів захисту інформації у навчальному закладі
2. Проектування системи захисту інформації
3. Побудова моделі захищеної мережі навчального закладу

Дата видачі завдання “1” листопада 2023 р.

Керівник магістерської роботи _____ / Волошин С.М. к.т.н., доцент /

(підпис)

(ПІБ, вчене звання і ступінь)

Завдання прийняв до виконання _____ / Сальник Д.С. /

(підпис)

(ПІБ)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка задачі магістерської роботи		Виконано
2	Аналіз предметної області		Виконано
3	Проектування системи		Виконано
4	Реалізація системи		Виконано
5	Тестування розробленої системи		Виконано
6	Оформлення пояснювальної записки		Виконано
7	Оформлення графічного матеріалу		Виконано

Студент _____ / Сальник Д.С. /
підпис ПІБ

Керівник проекту (роботи) _____ / Волошин С.М. /
підпис ПІБ

РЕФЕРАТ

Пояснювальна записка: 62с., 25 рис., 1 додаток, 31 використаних джерела.

СИСТЕМА, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, ШТУЧНИЙ ІНТЕЛЕКТ, БЕЗПЕКА, ІНФОРМАЦІЯ, РЕАЛЬНИЙ ЧАС, МОДЕЛІ ГЛИБОКОГО НАВЧАННЯ, OPENCV, TENSORFLOW, FLASK, PYTHON.

Мета роботи – створення комп'ютерної системи, здатної розпізнавати обличчя людей у реальному часі з використанням алгоритмів штучного інтелекту, що дозволить автоматизувати процес аутентифікації, підвищити безпеку та забезпечити швидкий і точний доступ до інформаційних ресурсів.

Об'єкт – система розпізнавання обличчя в реальному часі.

Предмет – алгоритми та програмне забезпечення для розпізнавання обличчя з використанням методів штучного інтелекту.

Робота складається з трьох розділів.

У першому розділі проаналізовано предметну область, розглянуті сучасні методи та алгоритми розпізнавання обличчя, а також тенденції розвитку штучного інтелекту у сфері безпеки.

У другому розділі висвітлено принципи та алгоритми роботи систем розпізнавання обличчя, зокрема моделі глибокого навчання, а також проаналізовано фреймворки та бібліотеки, такі як OpenCV і TensorFlow.

У третьому розділі детально описано процес створення системи розпізнавання обличчя, включаючи налаштування програм для обробки зображень у реальному часі.

В результаті виконання магістерської роботи проведено аналіз, розробка та тестування системи розпізнавання обличчя з використанням алгоритмів штучного інтелекту. Система може бути впроваджена для підвищення безпеки доступу до інформаційних систем і автоматизації процесів аутентифікації.

ЗМІСТ

РЕФЕРАТ	4
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Сучасні методи розпізнавання обличчя	8
1.1.1 Використання штучного інтелекту	11
1.1.2 Нейронні мережі та глибоке навчання для розпізнавання обличчя	12
1.2 Інструменти для реалізації систем розпізнавання обличчя	14
1.2.1 Бібліотеки для обробки зображень та відео	14
1.2.2 Фреймворки для глибокого навчання	16
1.2.3 Готові програмні рішення для розпізнавання обличчя	19
1.3 Проблеми впровадження систем розпізнавання обличчя в реальному часі	22
1.3.1 Вплив на продуктивність та ресурси	22
1.3.2 Проблеми з точністю через змінні умови	25
1.3.3 Етичні питання та захист приватності	26
2 РОЗРОБКА ТА ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧЧА, ВІКУ ТА СТАТТІ В РЕАЛЬНОМУ ЧАСІ	29
2.1 Вимоги до системи розпізнавання в реальному часі	29
2.2 Вибір алгоритмів і моделей для розпізнавання	31
2.3 Інтеграція зображень та їх обробка	34
2.4 Побудова та налаштування системи	37
2.4.1 Обробка відео-потoku за допомогою OpenCV	39
2.4.2 Використання моделей для визначення віку та статі з Caffe	41
2.4.3 Реалізація розпізнавання обличчя із використанням Face Recognition	43
3 ТЕСТУВАННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ	48
3.1 Тестування системи в реальному часі	48
3.2 Аналіз точності розпізнавання обличчя, визначення віку та статі	48
3.3 Тестування продуктивності при різних умовах освітлення	51

3.4 Використання різних зображень для оцінки моделі.....	53
ВИСНОВКИ.....	62
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТОК А.....	67

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

HOG — Вектор ознак гістограми орієнтованих градієнтів, Histogram of Oriented Gradients

ШІ — Штучний інтелект

CNN — Згорткові нейронні мережі, Convolutional Neural Networks

OpenCV — Бібліотека для виявлення об'єктів відкритим кодом, Open Source Computer Vision Library

LBP — Локальні бінарні шаблони, Local Binary Patterns

DNN — Глибокі нейронні мережі, Deep Neural Networks

SVM — Метод підтримки векторів, Support Vector Machine

API — Інтерфейс програмування додатків, Application Programming Interface

CPU — Центральний процесор, Central Processing Unit

GPU — Графічний процесор, Graphics Processing Unit

RAM — Оперативна пам'ять, Random Access Memory

HD — Висока роздільна здатність, High Definition

4K — Ультрависока роздільна здатність, Ultra High Definition (4K)

ГБ — Гігабайт

ВСТУП

Тема розпізнавання обличчя в реальному часі стала особливо актуальною з розвитком технологій штучного інтелекту. Швидке та точне визначення особи на основі зображень відіграє важливу роль в різних сферах, включаючи безпеку, маркетинг, охорону здоров'я та інші галузі. Від класичних методів розпізнавання обличчя до сучасних алгоритмів глибокого навчання, технологія пройшла довгий шлях еволюції, вдосконалюючись з кожним роком.

З розвитком штучного інтелекту та збільшенням обчислювальної потужності персональних комп'ютерів і мобільних пристроїв, можливості автоматичного розпізнавання обличчя вийшли на новий рівень. Висока точність та можливість роботи в реальному часі роблять ці системи все більш доступними для впровадження в реальні проекти. Використання технологій глибокого навчання та нейронних мереж дозволяє досягти високого рівня точності навіть при роботі зі зображеннями низької якості або за складних умов освітлення.

Важливість розпізнавання обличчя особливо зростає в контексті безпеки та автоматизації. Системи, що здатні розпізнавати осіб у реальному часі, можуть використовуватися для автоматичного контролю доступу на об'єкти, виявлення підозрілих дій, забезпечення безпеки в громадських місцях, а також в банківському секторі для захисту персональних даних користувачів. Однак, водночас із широкими можливостями таких систем, виникають питання щодо захисту приватності, етичності їх використання та правових аспектів.

Україна, як і інші країни, не стоїть осторонь цих глобальних тенденцій. В умовах посилення кіберзагроз та зростання цифрових ризиків, створення надійних систем розпізнавання обличчя, що здатні працювати в реальному часі, стає важливим завданням для багатьох компаній та державних установ. Впровадження таких систем в різні сфери життя сприяє підвищенню загального рівня безпеки.

Метою даної роботи є розробка комп'ютерної системи розпізнавання обличчя, яка здатна працювати в реальному часі, використовуючи технології

штучного інтелекту, зокрема інструменти OpenCV, TensorFlow та Caffe. Особливу увагу приділено інтеграції цієї системи з відео-потокком для обробки зображень, а також використанню відкритих платформ для забезпечення доступності та економічної ефективності розробленої системи.

В ході виконання роботи буде створено систему, що дозволить не тільки розпізнавати обличчя в режимі реального часу, але й визначати вік та стать людини, що може бути корисним для різних бізнес-задач, систем контролю доступу та забезпечення безпеки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сучасні методи розпізнавання обличчя

Розпізнавання обличчя це технологія, яка дозволяє автоматичним системам ідентифікувати або перевіряти особистість людини на основі візуальних характеристик її обличчя. Сучасні методи розпізнавання обличчя активно використовують новітні досягнення в галузі штучного інтелекту, машинного навчання, а також обробки та аналізу зображень. Технології стали настільки потужними, що розпізнавання може здійснюватися у режимі реального часу з високим рівнем точності, навіть у складних умовах, таких як зміни освітлення, кут зйомки або присутність сторонніх об'єктів [1].

Однією з ключових особливостей сучасних методів є здатність систем навчатися на великих масивах даних, що дозволяє підвищити ефективність і точність розпізнавання. Системи проходять тренування на зображеннях облич людей, аналізуючи унікальні риси обличчя, такі як відстань між очима, форму носа, підборіддя та інші анатомічні ознаки. Після цього, система може порівнювати нові зображення з тими, що зберігаються в базі даних, і визначати збіг.

Типовий процес розпізнавання обличчя, включає кілька етапів (рис. 1.1):

1. **Захоплення (Capturing):** На першому етапі система захоплює обличчя людини через камеру або інший засіб зйомки, наприклад, відеокамеру або мобільний пристрій. Камера може працювати в реальному часі або фіксувати зображення з попередньо записаного відео. Важливо, щоб зображення було достатньо якісним, оскільки чіткість та деталізація обличчя значно впливають на точність подальшого аналізу. На цьому етапі фіксуються первинні візуальні дані, такі як положення обличчя, розмір та орієнтація. У деяких випадках система може застосовувати попередню обробку зображення, наприклад, для покращення контрасту або усунення шумів;

2. Витягування (Extracting): Після захоплення зображення система переходить до етапу витягування ключових особливостей обличчя. Це найважливіший етап, оскільки саме на цьому кроці алгоритми аналізують обличчя та визначають його унікальні ознаки. Використовуються методи комп'ютерного зору, такі як виявлення точок на обличчі: форма очей, носа, губ, відстань між різними точками, а також інші морфологічні ознаки. Сучасні алгоритми глибокого навчання можуть також витягувати більш складні ознаки, які не завжди очевидні для людського ока. Це допомагає підвищити точність розпізнавання навіть у випадках, коли обличчя частково приховане або має різні вирази;

3. Порівняння (Comparing): На цьому етапі система порівнює отримані характеристики обличчя з шаблонами, збереженими в базі даних. Для цього використовуються спеціальні алгоритми, які зіставляють ключові особливості з відповідними характеристиками зареєстрованих користувачів. База даних може містити як фотографії облич, так і збережені векторні ознаки для прискорення процесу порівняння. Кожне обличчя представлено у вигляді набору унікальних параметрів, які забезпечують можливість розрізняти особистості навіть за наявності мінімальних відмінностей. Порівняння може відбуватися за допомогою евклідової відстані між векторами ознак або інших математичних методів;

4. Співставлення (Matching): Якщо система знаходить в базі даних обличчя, яке відповідає витягнутим характеристикам, вона ідентифікує особу. Цей процес відбувається на основі збігу між витягнутими ознаками обличчя та профілями, що зберігаються в базі даних. Якщо збіг є достатньо точним, система підтверджує особу як розпізнану. Якщо точний збіг відсутній або рівень подібності недостатній, система може визначити особу як нерозпізнану або запропонувати повторити аналіз. В залежності від налаштувань, система може або відразу припинити роботу, або спробувати покращити якість аналізу шляхом збирання додаткових даних. У сучасних

системах цей етап займає дуже мало часу завдяки потужним обчислювальним ресурсам та оптимізованим алгоритмам.

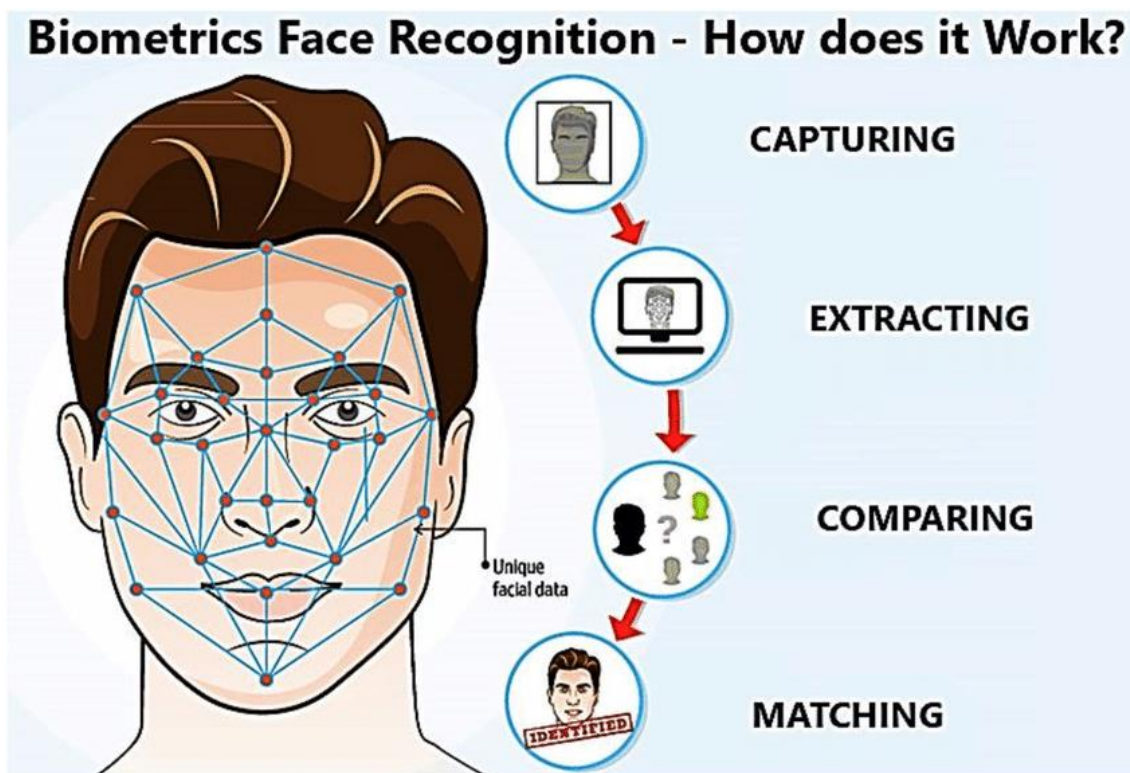


Рисунок 1.1 — Процес біометричного розпізнавання обличчя [2]

Цей процес є фундаментом для систем розпізнавання обличчя, які використовуються у різноманітних застосуваннях, включаючи системи безпеки, автоматичну ідентифікацію особистості, системи контролю доступу та інші застосування, де необхідно автоматично та точно визначати людину за її обличчям.

Сучасні технології розпізнавання обличчя не обмежуються статичними зображеннями, а здатні працювати з відео-потоками в реальному часі, що дозволяє використовувати їх у системах відеоспостереження, безпеки та авторизації [3]. Окрім ідентифікації, технології розпізнавання можуть також застосовуватися для аналізу віку, статі, емоційних станів, що розширює їхні можливості в комерційних і соціальних проектах. У системах розпізнавання обличчя використовуються різні підходи до аналізу та обробки зображень. Зокрема, поширеним є метод двоетапного розпізнавання: спочатку система знаходить обличчя на зображенні, а потім здійснює його ідентифікацію. Для цього широко застосовуються спеціальні

алгоритми для виявлення обличчя, такі як Histogram of Oriented Gradients (HOG), який аналізує текстуру зображення і визначає контури обличчя [4]. У свою чергу, для порівняння знайденого обличчя з наявними в базі використовується векторна репрезентація обличчя, де кожне обличчя представляється у вигляді числового вектора, що описує його унікальні характеристики.

1.1.1 Використання штучного інтелекту

Сучасні системи розпізнавання обличчя значно покращили свою точність і швидкість завдяки інтеграції алгоритмів штучного інтелекту (ШІ). ШІ дозволяє системам адаптуватися до різних умов, таких як зміни освітлення, кутів зйомки або якості зображень, що робить їх більш універсальними та ефективними.

Однією з ключових характеристик сучасних систем є їх багаторівнева архітектура (рис. 1.2).

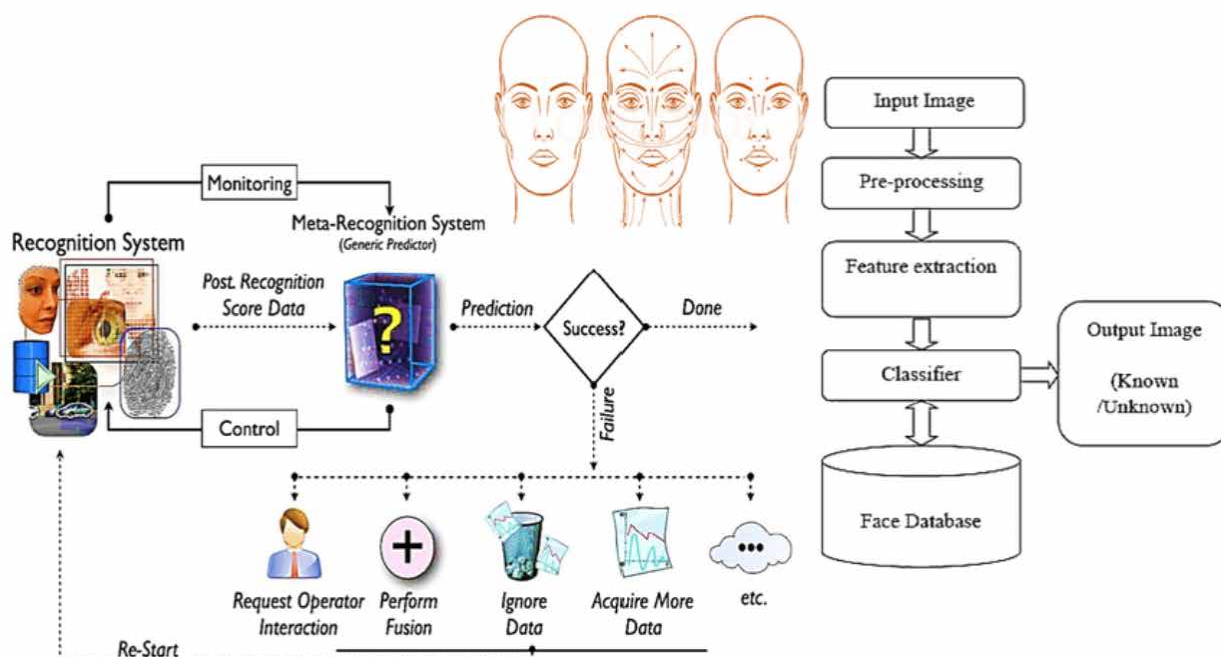


Рисунок 1.2 — Архітектура розпізнавання облич з використанням ШІ [5]

Процес розпізнавання обличчя починається з введення зображення (Input Image), яке проходить через етап попередньої обробки (Pre-processing). Це включає нормалізацію зображення та підготовку до подальшого аналізу. Після цього

виконується витягнення характеристик (Feature Extraction), де система визначає важливі риси обличчя, такі як відстань між очима, форму носа тощо. Наступним етапом є класифікатор (Classifier), який, на основі отриманих характеристик, порівнює їх із базою даних облич (Face Database). Якщо обличчя знайдене в базі, система класифікує його як відоме (Known), якщо ні як невідоме (Unknown).

У випадку успішного розпізнавання система може завершити процес, але якщо класифікація не вдалася, то запускається система Meta-Recognition, яка переглядає результати і робить прогноз успіху або невдачі розпізнавання. У разі невдачі система може здійснити додаткові дії, такі як збір нових даних або запит на взаємодію з оператором.

III відіграє ключову роль на всіх етапах розпізнавання обличчя, від виявлення на зображенні до порівняння отриманих результатів із наявними базами даних. Одна з основних переваг штучного інтелекту полягає у здатності систем самонавчатися, тобто покращувати свої результати з часом на основі попередніх даних. Це стає можливим завдяки алгоритмам машинного навчання, які дозволяють системі автоматично аналізувати великі масиви даних і виявляти в них закономірності.

1.1.2 Нейронні мережі та глибоке навчання для розпізнавання обличчя

Нейронні мережі, зокрема згорткові нейронні мережі (Convolutional Neural Networks, CNN), є основою для сучасних систем розпізнавання облич завдяки їхній здатності автоматично витягувати та аналізувати ключові риси обличчя на зображеннях. CNN використовують багатошарову архітектуру, де кожен шар аналізує зображення на різних рівнях: від базових контурів і текстур до складних структур, таких як форма носа, очей, рота та інших анатомічних характеристик. Це дає змогу мережі виявляти унікальні риси, необхідні для точного розпізнавання облич. Така архітектура дозволяє CNN навчатися на великих наборах даних і витягувати важливі візуальні ознаки, що робить їх ефективними у розпізнаванні навіть при зміні умов зйомки [6].

Глибоке навчання є методологією, що дозволяє мережам, таким як CNN, працювати з величезними наборами даних для тренування моделей розпізнавання. Глибоке навчання використовує велику кількість шарів у нейронних мережах для автоматичного навчання та вдосконалення з кожною новою ітерацією тренування. Це забезпечує можливість створення універсальних моделей, здатних розпізнавати обличчя навіть за складних умов: при зміні освітлення, ракурсу, або у випадках часткового закриття обличчя, наприклад окулярами чи масками. У дослідженнях зазначається, що глибоке навчання дозволяє системам покращувати свою точність на основі аналізу величезної кількості зображень з різними характеристиками [7-8].

Крім того, глибоке навчання забезпечує можливість розпізнавання облич у реальному часі, що є критично важливим для сфер, де швидкість та точність є ключовими факторами. Це робить системи розпізнавання облич на основі CNN надзвичайно корисними для завдань відеоспостереження, безпеки та контролю доступу. Завдяки здатності CNN масштабуватися, такі системи можуть одночасно обробляти кілька зображень або відео-потоків, що забезпечує високу продуктивність навіть при роботі з великим обсягом даних. Наприклад, у контексті систем безпеки та ідентифікації осіб, CNN можуть одночасно аналізувати обличчя кількох людей у кадрі, що дозволяє ідентифікувати осіб на багатолюдних заходах або у великих установах. Ключовою перевагою глибокого навчання є також його здатність до масштабування та адаптації. CNN можуть ефективно використовувати великі набори даних для створення моделей, які здатні узагальнювати інформацію та застосовувати її до нових, непередбачуваних ситуацій. Це робить системи розпізнавання обличчя на основі глибокого навчання дуже гнучкими, оскільки вони можуть швидко адаптуватися до нових даних і нових умов роботи, що дозволяє їм працювати на високому рівні продуктивності навіть у складних і динамічних умовах.

1.2 Інструменти для реалізації систем розпізнавання обличчя

Розпізнавання обличчя є складною багаторівневою задачею, яка вимагає використання спеціалізованих інструментів для обробки зображень, навчання моделей та інтеграції у реальні системи. Ці інструменти поділяються на кілька категорій, включаючи бібліотеки для обробки зображень і відео, фреймворки для глибокого навчання, а також готові програмні рішення.

1.2.1 Бібліотеки для обробки зображень та відео

Для створення систем розпізнавання обличчя важливо використовувати спеціалізовані бібліотеки, які дозволяють обробляти зображення, відео-потоки і виконувати базові операції з ними. Найбільш поширеними бібліотеками для таких задач є OpenCV, Dlib та Face Recognition. Кожна з них має свої унікальні можливості та використовується для різних завдань у сфері обробки зображень та відео.

OpenCV (Open Source Computer Vision Library) одна з найбільш популярних бібліотек для виявлення об'єктів та обробки зображень. OpenCV підтримує широкий спектр функцій, таких як обробка зображень, відео-потоків, розпізнавання об'єктів, обличчя, жестів, виявлення руху тощо [9]. Однією з ключових переваг OpenCV є підтримка багатьох мов програмування, зокрема Python, C++, Java та інші, що робить її доступною для широкого кола розробників. OpenCV також пропонує готові алгоритми для виявлення обличчя, такі як каскад Хаара, LBP, DNN та HOG в поєднанні з SVM. Каскад Хаара базується на використанні навченої моделі для швидкого виявлення обличчя на основі характерних ознак, таких як контури та тіні. Він простий у використанні та швидкий, проте чутливий до умов освітлення. LBP (Local Binary Patterns) використовує локальні текстурні патерни для розпізнавання обличчя, що робить його більш стійким до змін освітлення та контрасту в порівнянні з каскадом Хаара. DNN (Deep Neural Networks) застосовує нейронні мережі для виявлення облич,

використовуючи потужні попередньо навчені моделі, що забезпечує високу точність навіть у складних умовах, але потребує більше ресурсів. HOG + SVM обчислює градієнти напрямків у зображенні для виявлення обличчя і використовує метод підтримки векторів (SVM) для точнішої класифікації. Цей метод добре підходить для складних сценаріїв з багатьма об'єктами в кадрі [10-12].

Dlib це високоефективна бібліотека для машинного навчання та обробки зображень, яка стала відомою завдяки своїм можливостям у розпізнаванні облич [13]. Основною функцією Dlib у цій галузі є точна детекція облич на основі 68-точкової моделі ключових точок (facial landmarks). Ця модель дозволяє визначити певні унікальні орієнтири на обличчі. Вона використовує попередньо навчену модель для виявлення фіксованих точок на обличчі, які відповідають певним анатомічним структурам. У 68-точковій моделі кожна з 68 точок розташована на певних місцях обличчя, включаючи:

- 17 точок для окреслення лінії щелепи;
- 10 точок для контуру брів;
- 9 точок для форми носа;
- 12 точок для форми губ;
- 12 точок для контурів очей;
- 8 точок для окреслення зовнішнього контуру обличчя.

Ці точки використовуються для ідентифікації структури обличчя на зображеннях або відео, що робить можливим точне визначення розташування очей, носа, рота та інших важливих рис обличчя [14].

Face Recognition ще одна популярна бібліотека, що базується на Dlib, і вона спрощує процес розпізнавання облич завдяки простому API. Face Recognition дозволяє розробникам легко розпізнавати, ідентифікувати обличчя та здійснювати класифікацію, використовуючи готові моделі глибокого навчання. Однією з головних переваг бібліотеки є її простота в роботі, вона дозволяє виконувати складні завдання, як-то виявлення облич, визначення відповідності між обличчями або створення системи контролю доступу, без потреби в глибокому знанні алгоритмів машинного навчання. Face Recognition використовує HOG і нейронні

мережі для виявлення облич, що забезпечує високу точність при мінімальних зусиллях [15].

Таким чином, ці бібліотеки служать фундаментом для створення систем розпізнавання обличчя, забезпечуючи ефективну обробку візуальних даних. Вони дозволяють проводити попередню обробку зображень, аналізувати їх та готувати для подальшого використання у моделях машинного навчання та нейронних мережах.

1.2.2 Фреймворки для глибокого навчання

Фреймворки для глибокого навчання відіграють ключову роль у розробці та впровадженні систем розпізнавання обличчя. Вони надають інструменти та API для розробки нейронних мереж, їхнього тренування на великих масивах даних і тестування в різних сценаріях. Основними фреймворками для реалізації систем глибокого навчання є TensorFlow, Keras, PyTorch, MXNet та Caffe. Кожен з них має свої особливості, які дозволяють вирішувати різні завдання в галузі машинного навчання та штучного інтелекту.

TensorFlow один із найпоширеніших фреймворків для глибокого навчання, розроблений компанією Google. Він відомий своєю гнучкістю, масштабованістю та широким спектром інструментів для тренування нейронних мереж, включно з CNN, які є основою розпізнавання обличчя. TensorFlow підтримує як дослідницькі, так і комерційні проекти, пропонуючи можливість працювати з масивними наборами даних і використовувати потужності графічних процесорів для прискорення обчислень [16]. Окрім цього, TensorFlow має власний інтерфейс для візуалізації та аналізу процесу навчання моделей (рис. 1.3). Цей інтерфейс називається TensorBoard. Це потужний інструмент, який допомагає розробникам краще зрозуміти роботу нейронних мереж, відстежувати зміни у параметрах моделей, а також оцінювати їх ефективність під час тренувань. TensorBoard надає візуальні відображення процесу навчання, що дозволяє робити висновки про оптимізацію та налагодження моделі в реальному часі.

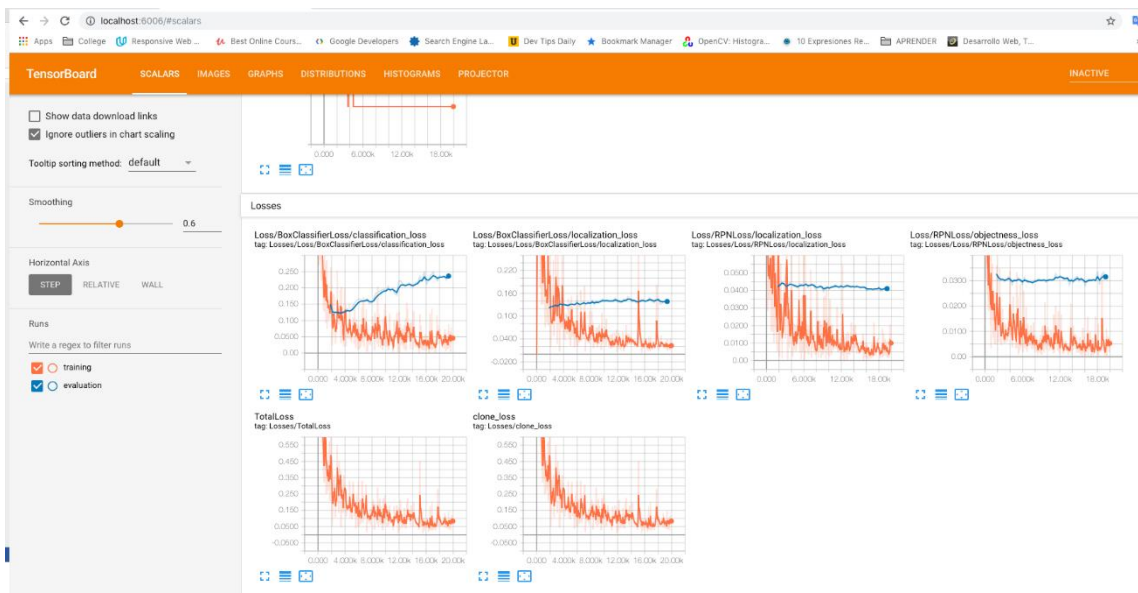


Рисунок 1.3 — Інтерфейс TensorBoard

Основні функції TensorBoard [17]:

- **Графи моделі:** TensorBoard дозволяє побудувати візуальний граф моделі, що демонструє архітектуру нейронної мережі та як кожен шар взаємодіє один з одним. Це допомагає легко побачити структуру та можливі проблеми у моделі;
- **Моніторинг метрик:** Під час тренування моделі TensorBoard відстежує і виводить графіки таких важливих метрик, як втрата (loss), точність (ассигасу), швидкість навчання та інші параметри. Це дає можливість розробникам оцінити, наскільки ефективно навчання проходить у кожному циклі (епосі), і чи потребує модель додаткової оптимізації;
- **Аналіз гістограм:** TensorBoard дозволяє аналізувати розподіл ваг і активацій для кожного шару мережі. Це корисно для визначення того, як ваги змінюються під час навчання, і чи виникають проблеми, такі як перенавчання (overfitting) або недостатнє навчання (underfitting);
- **Візуалізація вбудовань (embeddings):** TensorBoard може візуалізувати вбудовування (наприклад, у випадку обробки текстів), що дозволяє наочно побачити, як модель представляє різні типи даних у багатовимірному просторі;

- Аналіз часу тренування: TensorBoard також дозволяє вивчати, скільки часу займає тренування на кожному кроці, що допомагає оцінити продуктивність моделі та обчислювальну ефективність.

TensorBoard надає зручні інструменти для комплексного моніторингу та аналізу моделей під час їх навчання, що значно полегшує процес оптимізації нейронних мереж у TensorFlow.

Keras високорівневий фреймворк для розробки нейронних мереж, який працює на основі TensorFlow, Theano або інших серверних систем. Його головна мета, спростити процес створення та тренування нейронних мереж завдяки зрозумілому та інтуїтивному API [18]. Keras дозволяє швидко створювати та тестувати різні архітектури мереж, що робить його ідеальним вибором для новачків у глибокому навчанні та тих, хто хоче швидко побудувати прототип своєї моделі.

PyTorch потужний фреймворк для глибокого навчання, розроблений компанією Facebook AI Research. Він відомий своєю динамічною обчислювальною графікою, що дозволяє легко змінювати архітектуру мережі під час виконання програми [19]. PyTorch є популярним серед дослідників і розробників через гнучкість і простоту в роботі з різними нейронними мережами. Крім того, PyTorch пропонує модульну систему для тренування моделей, що дозволяє краще контролювати процес навчання та налаштовувати кожен етап.

MXNet це інший популярний фреймворк для глибокого навчання, розроблений Apache Software Foundation. Його особливість полягає в підтримці обчислень на кількох графічних процесорах і можливості легкої інтеграції в хмарні сервіси, такі як Amazon Web Services. MXNet є особливо привабливим для компаній, які займаються великими обсягами даних, і тих, хто використовує хмарні обчислення. Він також підтримує кілька мов програмування, таких як Python, C++, Julia та інші [20].

Caffe спеціалізований фреймворк для глибокого навчання, який спочатку був розроблений для комп'ютерного зору. Він став популярним завдяки своїй швидкості та ефективності при роботі з CNN, що робить його ідеальним для систем розпізнавання обличчя. Caffe відомий своєю високою продуктивністю при роботі з

великими наборами даних і можливістю працювати на графічних процесорах, що значно прискорює процес навчання. Однак, порівняно з TensorFlow або PyTorch, Caffe має менш гнучку архітектуру, що робить його придатним для специфічних застосувань, таких як розпізнавання зображень.

Кожен з цих фреймворків має свої унікальні можливості, які дозволяють обирати найбільш підходящий інструмент залежно від специфіки завдання в розпізнаванні облич або обробці зображень.

1.2.3 Готові програмні рішення для розпізнавання обличчя

Аналіз готових рішень для розпізнавання обличчя демонструє, що існує низка потужних API та інструментів, які надають розробникам можливість швидко інтегрувати технології розпізнавання у свої проєкти без необхідності самостійної розробки моделей. Ці системи базуються на передових алгоритмах глибокого навчання, що забезпечує високу точність розпізнавання. Однак, при виборі такого рішення важливо враховувати не лише його переваги, але й недоліки, щоб зрозуміти, наскільки ефективно воно відповідає вимогам конкретного завдання.

FaceNet відкрита модель від Google, яка використовує векторну репрезентацію облич для порівняння схожості. FaceNet генерує компактні векторні представлення облич, що дозволяє ефективно визначати подібність між різними зображеннями. Важливою перевагою FaceNet є те, що вона може бути налаштована на класифікацію та кластеризацію, що робить її універсальною для різних завдань. FaceNet досягає високої точності навіть на складних наборах даних, перевищуючи багато інших сучасних моделей. Однак недоліком залишається висока вимога до обчислювальних ресурсів, особливо під час обробки великих обсягів даних. Це може стати проблемою для проєктів, які не мають доступу до потужних серверів або кластерів для обчислень. Інтеграція FaceNet також може вимагати налаштувань, таких як коректне підготування даних та налаштування нейронної мережі для використання на різних платформах, що може бути складним для новачків [21].

DeerFace модель від Facebook, яка використовує глибокі нейронні мережі для розпізнавання облич, досягаючи дуже високого рівня точності. Вона була однією з перших моделей, які використовували глибокі нейронні мережі для обробки обличчя, і зараз залишається одним із стандартів у цій сфері. Перевагою DeerFace є її стійкість до зміни освітлення та кута огляду, що дозволяє працювати з обличчями у різних положеннях. Водночас, DeerFace вимагає значних обчислювальних ресурсів для тренування та використання, що може зробити її менш доступною для невеликих команд. Крім того, через обмежену документацію, інтеграція може викликати труднощі, особливо для розробників без досвіду в роботі з нейронними мережами. Ще одним недоліком є обмежена можливість налаштування, система призначена для загального використання і не дозволяє легко адаптувати моделі під специфічні вимоги проєктів [22].

Microsoft Face API комерційний продукт, що надає великий набір функцій для аналізу та розпізнавання облич, таких як аналіз емоцій, визначення віку, статі, а також ідентифікація осіб. Однією з головних переваг є легкість інтеграції, адже Microsoft Face API має добре задокументоване API, що дозволяє швидко налаштовувати системи для різних додатків. Однак залежність від Microsoft Azure може призвести до значних фінансових витрат, особливо для стартапів та невеликих проєктів, оскільки використання сервісів Azure вимагає підписки. Крім того, користувачі можуть бути залежними від зовнішніх серверів для обробки даних, що підвищує ризик затримок або відсутності доступу у випадку перебоїв у роботі сервісів. Це може стати серйозною проблемою для проєктів, які критично залежні від безперервної роботи системи [23].

Google Cloud Vision один із провідних сервісів для розпізнавання облич та інших завдань комп'ютерного зору, що пропонується в рамках хмарної платформи Google. Окрім розпізнавання облич, Cloud Vision API може виконувати такі завдання, як аналіз емоцій, визначення статі та віку, а також розпізнавання об'єктів. Однією з основних переваг є легкість у використанні та висока точність, оскільки Google постійно оновлює свої алгоритми. Проте, основний недолік сервісу полягає в необхідності підписки на хмарні сервіси Google, що може бути фінансово

обтяжливим для проєктів з обмеженим бюджетом. Залежність від інтернет-з'єднання також означає, що якість та швидкість обробки запитів можуть залежати від стабільності мережі, що може стати критичним фактором для великих або швидких систем. Крім того, можливі затримки при обробці запитів можуть вплинути на загальну продуктивність системи, особливо під час високих навантажень [24].

Незважаючи на високу точність та функціональність, більшість готових рішень мають такі обмеження, як високі витрати на підписки, залежність від хмарних сервісів, затримки у роботі з великими обсягами даних і складність інтеграції. Крім того, ці рішення часто обмежують можливість гнучкого налаштування під специфічні вимоги проєкту, що може бути критичним для деяких компаній або стартапів, які прагнуть створити унікальні рішення.

Розроблена система має низку переваг перед згаданими аналогами. Вона не лише розпізнає обличчя, але й аналізує вік та стать користувача, що розширює її функціональні можливості та робить її корисною для різних застосувань, таких як контроль доступу, маркетингові дослідження чи навіть системи спостереження. Система працює автономно, без необхідності підключення до інтернету або зовнішніх хмарних платформ, що знижує затримки у роботі та покращує швидкість обробки запитів. Після запуску система відкриває камеру та автоматично перевіряє обличчя користувача. Якщо обличчя вже збережене у базі даних (теці з обличчями), система успішно ідентифікує користувача, надаючи інформацію про його вік і стать. У випадку, коли обличчя не знайдено в базі, система повідомляє про його невиявлення, але все одно може визначити вік і стать, що робить її корисною навіть без попередньої реєстрації користувачів.

Цей підхід дозволяє здійснювати швидке розпізнавання без необхідності підключення до хмарних сервісів чи використання комерційних підписок, роблячи систему ефективною та доступною для локальних задач. Окрім того, така система забезпечує підвищену конфіденційність, оскільки дані зберігаються локально, що може бути важливим для організацій, які прагнуть уникнути передачі чутливих даних у хмару.

1.3 Проблеми впровадження систем розпізнавання обличчя в реальному часі

Системи розпізнавання обличчя в реальному часі є потужним інструментом для забезпечення безпеки та ідентифікації, проте їх впровадження стикається з низкою проблем. Ці проблеми можуть впливати як на технічні аспекти, такі як продуктивність і точність, так і на соціальні та етичні питання.

1.3.1 Вплив на продуктивність та ресурси

Розпізнавання обличчя в реальному часі це одна з найбільш ресурсомістких задач, що потребує комплексної взаємодії різних компонентів системи, таких як центральний процесор (CPU), графічний процесор (GPU) та оперативна пам'ять (RAM). Ефективність роботи таких систем багато в чому залежить від швидкості та продуктивності цих компонентів, оскільки в процесі обробки відео-потoku необхідно аналізувати та порівнювати велику кількість даних у кожному кадрі [25-26].

CPU є «серцем» будь-якої системи розпізнавання обличчя, оскільки він відповідає за виконання алгоритмів виявлення та ідентифікації. Процесор обробляє кожен кадр відео-потoku та здійснює попередню обробку даних: перетворює зображення у придатний для аналізу формат, керує алгоритмами розпізнавання, а також проводить порівняння обличч із базою даних. Окрім цього, CPU відповідає за координацію всієї системи, починаючи від захоплення відео-потoku, аналізу даних і до фінального результату розпізнавання, який система виводить на екран. Чим більше обличч система повинна обробити на кадр і чим більше кадрів надходить щосекунди, тим більше навантаження припадає на процесор. У системах реального часу це особливо критично, адже користувачі очікують негайної реакції без помітних затримок. Для підтримки високої швидкості обробки, особливо при роботі з великими обсягами відеоданих, потрібен багатоядерний процесор із високою тактовою частотою, який дозволяє виконувати паралельну обробку різних задач.

Недостатня продуктивність процесора може призвести до затримок, падіння частоти кадрів або навіть до збоїв у роботі системи, особливо при обробці великих відеопотоків або коли необхідно одночасно розпізнавати кілька облич. Крім того, надмірне навантаження на процесор може негативно вплинути на стабільність системи, збільшивши її енергоспоживання та температурний режим, що потребуватиме додаткових засобів охолодження і може зменшити загальну тривалість роботи системи.

GPU є ключовим компонентом у сучасних системах розпізнавання обличчя, оскільки він значно прискорює обробку відео-потoku завдяки можливості паралельного виконання великої кількості операцій одночасно. Графічні процесори оптимізовані для роботи з нейронними мережами, які вимагають інтенсивних обчислень на основі матричних операцій, таких як CNN, які часто використовуються для розпізнавання обличчя. Завдяки GPU система може швидко аналізувати та обробляти великі обсяги даних, забезпечуючи високу продуктивність у режимі реального часу. Це особливо важливо для додатків, де критичною є швидкість реакції, наприклад, в системах відеоспостереження, безпеки, контролю доступу, або для персоналізованих рекламних платформ. Використання потужних графічних процесорів, таких як NVIDIA RTX або Tesla, дозволяє системам досягати виняткової точності та швидкості обробки без значних затримок. Однак такі рішення можуть бути досить дорогими, і вони часто вимагають значного енергоспоживання, що може бути проблемою для невеликих проектів або для пристроїв, які працюють в автономному режимі. Незважаючи на це, GPU суттєво підвищують загальну ефективність і швидкість розпізнавання, роблячи такі системи більш практичними та надійними для широкого використання.

RAM відіграє вирішальну роль у забезпеченні стабільної роботи системи розпізнавання обличчя, оскільки вона відповідає за зберігання та обробку тимчасових даних, які використовуються для виконання алгоритмів у режимі реального часу. Оперативна пам'ять дозволяє швидко зберігати та обробляти кадри відео-потoku, що надходять із камери, забезпечуючи безперебійну роботу системи.

У випадках, коли система обробляє відео з високою частотою кадрів або у високій роздільній здатності (HD, 4K), оперативна пам'ять споживається ще активніше. Особливо це стосується ситуацій, коли одночасно обробляється кілька відео-потоків або коли система працює з великими базами даних облич. Для забезпечення швидкої та плавної обробки даних у таких умовах системам може знадобитися від 8 до 32 ГБ оперативної пам'яті, залежно від обсягу та якості даних, що обробляються. Якщо оперативної пам'яті недостатньо, це може призвести до затримок у відображенні або аналізі кадрів, або навіть до «вильоту» програми, що значно впливає на стабільність роботи системи.

Окрім оперативної пам'яті, важливим аспектом є наявність достатнього дискового простору для зберігання відеоданих, особливо у випадках, коли система працює з великими обсягами даних у високій якості, такими як HD або 4K. Швидкий доступ до дискового простору та ефективне управління цими даними є важливими для підтримки безперебійної роботи системи, особливо коли вона використовує локальне зберігання. Якщо система використовує хмарні сервіси для зберігання або обробки даних, пропускну здатність мережі також стає критичним фактором. У таких випадках повільне інтернет-з'єднання може призвести до затримок у передачі відео або даних для подальшого аналізу, що вплине на загальну продуктивність системи.

Постійна обробка відео-потоків створює постійне навантаження на всі компоненти системи. Кількість облич, які система має обробити у кожному кадрі, також впливає на рівень навантаження. Якщо на кадрі одночасно з'являються десятки або навіть сотні облич, система повинна обробити кожне окремо, що вимагає додаткових обчислювальних ресурсів. У випадку систем, що використовують нейронні мережі для розпізнавання облич, процес навчання може тривати кілька днів або тижнів, залежно від розміру наборів даних та потужності обладнання.

Загалом, ефективне впровадження систем розпізнавання обличчя в реальному часі потребує ретельного планування ресурсів, зокрема, вибору

оптимальних процесорів, графічних карт та обсягу оперативної пам'яті для забезпечення стабільної та швидкої роботи системи.

1.3.2 Проблеми з точністю через змінні умови

Одним із найбільших викликів для систем розпізнавання обличчя є точність їхньої роботи в умовах, що змінюються. Такі системи можуть показувати високі результати у стабільних умовах, але їхня ефективність може різко знижуватися при зміні освітлення, кута зйомки, якості зображення та інших факторів [27-28]. У таблиці 1.1 представлені основні змінні умови та їхній вплив на точність розпізнавання.

Таблиця 1.1 — Основні змінні умови

Змінна умова	Опис	Вплив на розпізнавання
Освітлення	Недостатнє або надмірне освітлення, нерівномірне освітлення	Сильно знижує точність. Може викликати труднощі у визначенні контурів обличчя та деталей рис.
Кут зйомки	Зміна кута зйомки, особливо якщо обличчя не фронтальне	Складно розпізнати частково або повністю обернені обличчя. Система може не знайти ключові точки для ідентифікації.
Якість зображення	Низька роздільна здатність,	Низька роздільна здатність робить неможливим розпізнавання ключових деталей, що сильно знижує точність.
Перешкоди на обличчі	Носіння окулярів, масок, головних уборів	Складність у визначенні повного обличчя, система може помилково класифікувати або взагалі не розпізнати обличчя.
Вікові зміни	Зміни обличчя з віком або наявність косметики	Система може не розпізнати особу через відмінності між зображенням у базі даних та новим виглядом особи.
Емоційні вирази	Сильно змінені вирази обличчя, такі як посмішка або гнів	Впливає на точність ідентифікації, оскільки система може не розпізнати змінену форму обличчя.

Шум у відео-потоці	Зовнішні фактори, такі як тіні, контраст або нерівномірність відеосигналу	Викликає складності у виявленні облич, що може призводити до помилкових позитивних результатів або відсутності розпізнавання.
Фонові елементи	Насичений фон або велика кількість деталей за обличчям	Фонові елементи можуть створювати перешкоди для системи у виявленні облич, особливо в динамічних кадрах.

Як видно з таблиці, змінні умови можуть суттєво вплинути на точність роботи систем розпізнавання обличчя. Системи, які працюють в реальному часі, потребують більш точного алгоритмічного налаштування для забезпечення стабільної роботи в різних умовах. Одним із можливих шляхів покращення може бути використання алгоритмів машинного навчання для адаптації до умов, що змінюються, таких як зміни освітлення або кута зйомки. Ретельне налаштування та оптимізація параметрів системи дозволить досягти високої точності навіть за складних умов експлуатації.

1.3.3 Етичні питання та захист приватності

Системи розпізнавання обличчя, які використовуються в реальному часі, дійсно викликають численні етичні та правові питання, особливо в контексті захисту приватності і прав людини.

Одним із ключових аспектів є згода на використання даних. Часто такі системи впроваджуються без чіткої згоди від осіб, які піддаються розпізнаванню, що викликає питання про правомірність використання технологій у публічних місцях. Закон України «Про захист персональних даних» [30] регулює питання отримання згоди, проте біометричні дані, такі як зображення обличчя, потребують особливої уваги, оскільки їх використання може мати серйозні наслідки для приватності громадян.

Зберігання та використання біометричних даних також є критичним питанням. Оскільки ці дані є надзвичайно чутливими, недостатній контроль за

їхнім зберіганням або захистом може призвести до їх витоку, що створює ризики зловживань з боку зловмисників. Ці дані можуть бути використані у кібератаках, що підкреслює необхідність ретельного захисту інформаційних систем, які використовують біометричну інформацію.

Неправомірне використання таких систем викликає додаткові занепокоєння. Використання технологій для масового відстеження або контролю без належних правових обмежень може призвести до порушень прав людини. Це може стосуватися і порушення прав на приватність, коли системи використовуються для негласного спостереження за особами без їхньої згоди або на основі дискримінаційних ознак.

Ще одним важливим аспектом є точність алгоритмів розпізнавання облич. Деякі з них можуть мати упередження, що відображається у менш точній роботі з певними етнічними або віковими групами. Це може призводити до дискримінації або неправильної ідентифікації особи, що, у свою чергу, може мати серйозні соціальні наслідки. Наприклад, неправильно ідентифікована особа може зазнати необґрунтованих переслідувань або інших обмежень.

Використання таких систем у правоохоронних органах стає дедалі популярнішим, але також викликає занепокоєння щодо прав громадян. Масове впровадження технологій для спостереження за громадянами може порушувати їхні права на приватність і призводити до зловживань, якщо правові рамки не забезпечують належний контроль за їх використанням. Чітке регулювання цього питання є необхідним для забезпечення балансу між безпекою та правами людини.

Захист від кібератак є невід'ємною частиною роботи з біометричними даними. Кіберзагрози становлять реальну небезпеку для систем, які обробляють чутливу інформацію, і будь-який витік таких даних може мати катастрофічні наслідки для особистої безпеки користувачів. Тому впровадження сильних захисних механізмів є обов'язковою умовою для забезпечення належного рівня кібербезпеки в системах розпізнавання обличчя.

Етичні та правові аспекти є надзвичайно важливими під час впровадження систем розпізнавання обличчя. Важливо забезпечити, щоб такі системи

використовувалися відповідально, з належним захистом приватності та дотриманням етичних стандартів. Прозорість у використанні біометричних даних, обмеження на їх зберігання та обробку, а також чіткі межі застосування технології розпізнавання обличчя мають бути на першому місці для мінімізації ризиків зловживання та порушень прав людини.

2 РОЗРОБКА ТА ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧЧА, ВІКУ ТА СТАТТІ В РЕАЛЬНОМУ ЧАСІ

2.1 Вимоги до системи розпізнавання в реальному часі

Системи розпізнавання обличчя в реальному часі повинні відповідати низці вимог, щоб забезпечити ефективну та точну роботу. Вимоги являють собою набір умов або функціональних можливостей, які повинні бути реалізовані в системі, щоб вона відповідала очікуванням користувачів і забезпечувала необхідну продуктивність. Вимоги можуть бути як функціональними так і нефункціональними. Функціональні вимоги описують, які конкретні дії повинна виконувати система для досягнення своїх цілей. Вони визначають, що саме система робить, як вона реагує на взаємодію користувача, та які задачі вирішує, їх наведено в таблиці 2.1.

Таблиця 2.1 — Основні функціональні вимоги до системи

Вимога	Опис
Виявлення обличчя на зображенні	Система повинна вміти ідентифікувати обличчя на завантажених зображеннях або з відео-потoku.
Розпізнавання особи	Якщо обличчя знайдено, система повинна порівняти його з наявними зображеннями у базі даних (папці) і визначити, чи є відповідність.
Аналіз віку	Система повинна визначити вік людини на основі її обличчя.
Аналіз статі	Система повинна визначити стать людини, використовуючи моделі для розпізнавання гендерних характеристик.
Підтримка завантаження фото	Користувач має мати можливість завантажувати фото в систему для подальшого аналізу.
Виведення результатів	Система повинна відображати результат розпізнавання на екрані, додаючи ім'я до фото або показуючи повідомлення «Не виявлено».

Продовження таблиці 2.1

Пошук у базі даних	Система повинна шукати відповідність обличчя у папці з уже завантаженими фото і визначати, чи є фото відповідної людини.
Використання моделей для розпізнавання	Система повинна використовувати попередньо навчені моделі для аналізу обличчя та визначення основних характеристик (стать, вік).
Обробка відео-потoku	Система повинна мати можливість обробляти кадри з відео-потoku в реальному часі.
Відображення рамки навколо обличчя	Система повинна показувати рамку навколо розпізнаного обличчя, з зазначенням віку та статі, якщо відповідність знайдено то ще й імені.

Нефункціональні вимоги визначають обмеження та критерії якості, за якими оцінюється робота системи. Вони включають вимоги до швидкості, продуктивності, безпеки та масштабованості, їх наведено в таблиці 2.2.

Таблиця 2.2 — Основні нефункціональні вимоги до системи

Вимога	Опис
Швидкість обробки	Система повинна виконувати розпізнавання обличчя в реальному часі з мінімальними затримками.
Масштабованість	Система повинна бути здатна обробляти збільшену кількість облич або відео-потоків без втрати продуктивності.
Надійність	Система повинна забезпечувати надійну роботу без збоїв під час роботи з великими обсягами даних або тривалого використання.
Безпека	Дані користувача (фото та результати розпізнавання) повинні зберігатися безпечно та не передаватися стороннім особам без дозволу.
Вимоги до обладнання	Система повинна мати мінімальні вимоги до обладнання, для забезпечення стабільної роботи.

Зручність у використанні	Інтерфейс системи має бути інтуїтивно зрозумілим і не вимагати додаткових знань від користувача.
Обробка помилок	Система повинна адекватно обробляти можливі помилки, такі як неможливість розпізнавання обличчя або завантаження некоректного зображення.
Оптимізація використання ресурсів	Система повинна ефективно використовувати обчислювальні ресурси для зменшення навантаження на процесор та графічний процесор.

Вимоги до системи розпізнавання обличчя в реальному часі охоплюють як функціональні, так і нефункціональні аспекти. Функціональні вимоги забезпечують правильну роботу системи, включаючи розпізнавання обличчя, аналіз віку та статі, а також роботу з базою даних для порівняння зображень. Нефункціональні вимоги гарантують якісну роботу системи, зокрема швидкість обробки, надійність, безпеку та масштабованість. Впровадження таких вимог дозволяє створити ефективну систему, яка відповідає потребам користувачів і виконує свої задачі з високою точністю та продуктивністю.

2.2 Вибір алгоритмів і моделей для розпізнавання

У сучасних системах розпізнавання широко використовуються алгоритми глибокого навчання та нейронні мережі, які забезпечують високу точність навіть в умовах зміни освітлення, кутів зйомки та інших факторів. Вибір алгоритмів та моделей залежить від багатьох параметрів, включаючи швидкість обробки, точність і доступність ресурсів для обчислень. Для розпізнавання обличчя обрано бібліотеку `face_recognition`, яка базується на сучасних методах обробки зображень. Вона використовує попередньо натреновані нейронні мережі для кодування облич у векторні представлення та подальшого порівняння облич у просторі ознак.

Основна перевага цієї бібліотеки полягає в її простоті використання та високій точності. Вона дозволяє швидко налаштувати систему для порівняння облич без потреби у власній нейронній мережі для навчання, що суттєво зменшує час на розробку системи.

Для визначення віку та статі використовуються моделі, побудовані на основі CNN. Ці моделі були завантажені з попередньо натренованими параметрами у форматі Caffe [31]:

- Gender_net використовується для визначення статі (чоловік або жінка);
- Age_net використовується для визначення віку людини на основі обличчя. Вік класифікується на певні вікові групи, що дозволяє ефективно оцінити вікову категорію особи.

Ці моделі відомі своєю точністю та швидкістю, а їх інтеграція в систему дозволяє в реальному часі обробляти відео-потоки та надавати результати без значних затримок.

Алгоритм розпізнавання обличчя, що використовується в системі, базується на технології векторизації облич або «face encodings», яка дозволяє перетворювати зображення обличчя на числові вектори, що описують унікальні риси кожної людини.

Спочатку алгоритм шукає обличчя на зображенні або у відео-потоці. Для цього використовуються спеціальні алгоритми виявлення облич, наприклад, алгоритми на основі HOG або глибоких нейронних мереж. Алгоритм шукає характерні риси обличчя, такі як очі, ніс, рот і контури обличчя, щоб точно визначити положення облич на зображенні. Після виявлення облич на зображенні, алгоритм генерує векторні представлення для кожного обличчя. Цей процес включає перетворення обличчя на вектор із заздалегідь визначеною кількістю чисел, які представляють ключові риси обличчя, такі як форма очей, носа, підборіддя тощо. Такі вектори мають фіксовану кількість вимірів (наприклад, 128 або 256), і вони є унікальними для кожної людини.

Згенеровані вектори для кожного нового обличчя порівнюються з базою векторів (face encodings), які були попередньо згенеровані та збережені для вже

відомих осіб. Векторні представлення для збережених обличч завантажуються з папки «images». Для порівняння використовується евклідовий або косинусний метод визначення схожості між векторами. Якщо вектор нового обличчя близький до вектора зі збереженої бази, це означає, що система знайшла відповідну особу.

Кожен вектор порівнюється зі всіма векторами в базі, і обчислюється відстань між ними. Якщо відстань між векторами менша за певний поріг, вважається, що обличчя було розпізнано, і система повертає ім'я особи, відповідної цьому обличчю. Якщо жоден із векторів у базі не відповідає вектору нового обличчя, система повідомляє, що обличчя не виявлено. Це може означати, що таке обличчя раніше не було додане до бази даних або є невідомим для системи.

Виходячи з опису алгоритму архітектуру системи можна представити у вигляді діаграми (рис. 2.1).

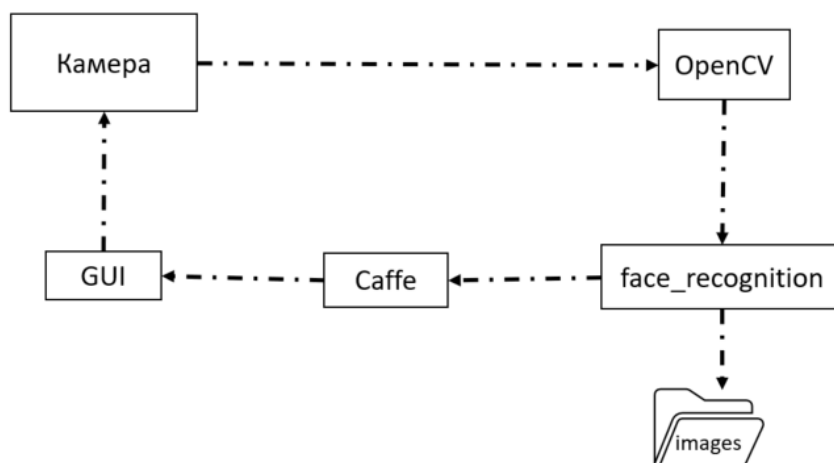


Рисунок 2.1 — Діаграма архітектури системи

Компоненти системи:

- Камера: Збирає відео-потік, який надсилається для подальшої обробки;
- Модуль обробки відео (OpenCV): Обробляє відео-потік, перетворює його на кадри для подальшого розпізнавання обличчя;

- Модуль розпізнавання обличчя (face_recognition): Виявляє обличчя на кожному кадрі та порівнює їх з базою даних зображень, щоб визначити, чи було знайдено відповідність;
- Модуль визначення віку та статі (Caffe): Використовує моделі для визначення віку та статі виявленого обличчя;
- База даних облич (тека images): Зберігає зображення облич для порівняння з тими, що були виявлені в кадрі;
- Інтерфейс користувача (GUI): Відображає результат обробки – ім'я користувача, стать, вік або, якщо обличчя не було знайдено то замість імені буде надпис «Не виявлено».

2.3 Інтеграція зображень та їх обробка

У системі розпізнавання обличчя зображення відіграють ключову роль, оскільки вони є основним джерелом даних для аналізу та порівняння. Вони використовуються для навчання системи та подальшого розпізнавання облич у реальному часі. Для кожного зображення система генерує унікальні векторні представлення, які потім порівнюються з векторними представленнями нових облич у відео-потоці. Зображення осіб завантажуються у теку «images» (рис. 2.2), і кожне з них представляє певну особу. Ім'я зображення відповідає імені людини на фото.

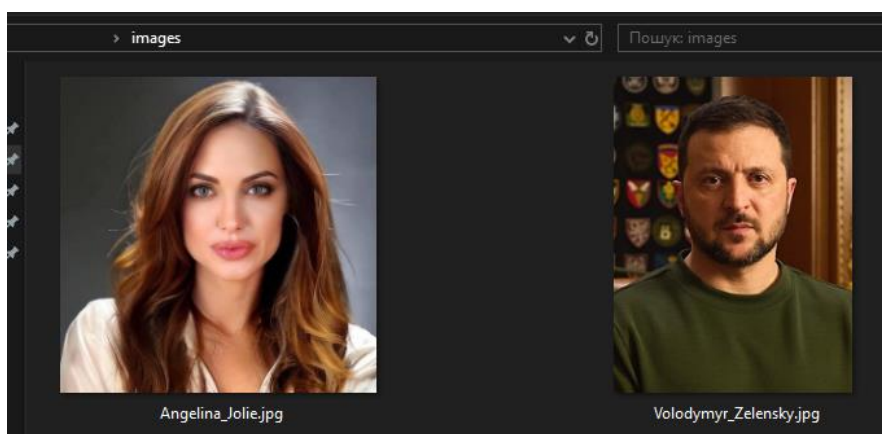


Рисунок 2.2 — Тека «images»

Ці зображення кодуються у вектори, які пізніше використовуються для порівняння з новими обличчями у реальному часі. Під час роботи система захоплює обличчя з камери у реальному часі і порівнює їх з уже збереженими векторами облич із зображень (рис. 2.3).

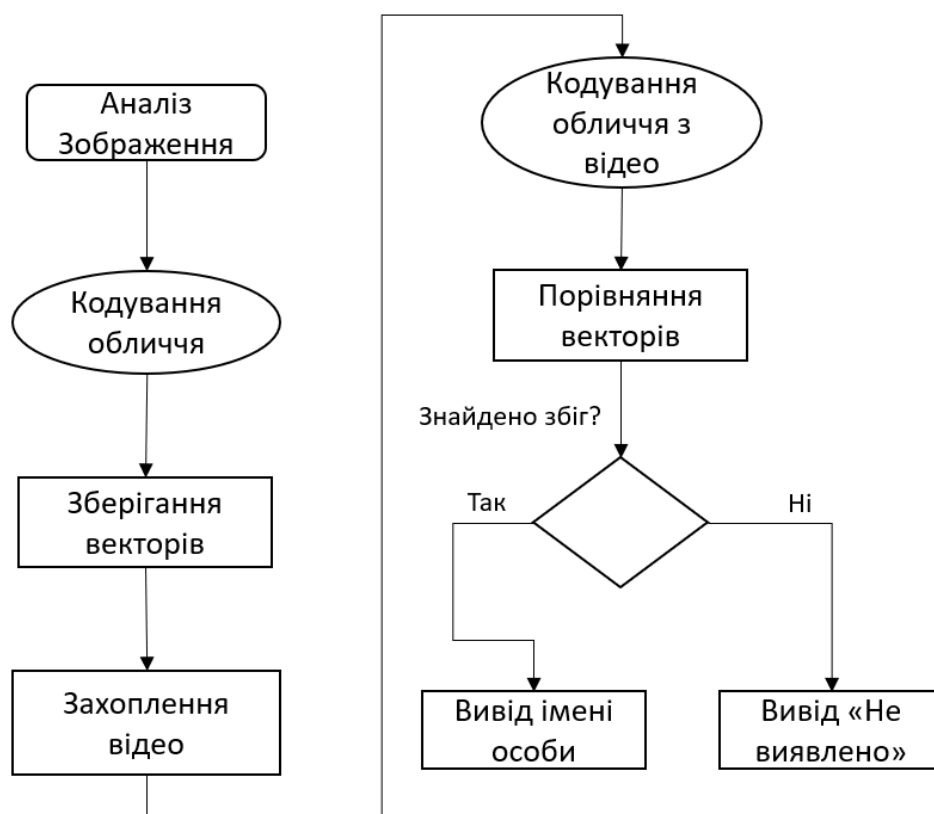


Рисунок 2.3 — Блок-схема обробки зображення системою

Блок-схема демонструє основні кроки обробки зображень та відео-потоків у розробленій системі розпізнавання обличчя. Детальний опис кожного етапу підкреслює ключові аспекти роботи системи:

1. Аналіз зображення: На початковому етапі система отримує доступ до зображень, які завантажуються з теки. Це первинний процес, де система отримує вихідні дані для подальшого аналізу та порівняння. Якість зображень, їх чіткість і розмір мають безпосередній вплив на точність подальшого розпізнавання;

2. Кодування обличчя: Після отримання зображення система здійснює його аналіз, витягаючи унікальні риси обличчя. Ці риси перетворюються на векторну репрезентацію, яка містить усю необхідну інформацію для порівняння. Кодування включає ключові характеристики, такі як відстань між очима, форма обличчя, розташування рис, що дозволяє системі точно визначати особу;
 3. Зберігання векторів: Після кодування обличчя векторні представлення зберігаються в пам'яті системи. Це важливий етап, оскільки від якості й точності збережених векторів залежить швидкість і ефективність майбутніх порівнянь. Дані вектори зберігаються для миттєвого доступу під час обробки нових зображень та відео;
 4. Захоплення відео: Система починає захоплення відео-потoku в режимі реального часу за допомогою камери. Відео-потік служить джерелом нових даних, де виявляються обличчя для подальшого аналізу та порівняння із вже збереженими зображеннями;
 5. Кодування обличчя з відео: Коли система виявляє обличчя у відео-потoci, вона кодує їх у векторні представлення так само, як і для зображень. Цей процес відбувається в реальному часі, що вимагає високої продуктивності обчислювальних ресурсів для збереження точності при високій швидкості обробки;
 6. Порівняння векторів: Після отримання векторних представлень нових обличчя система порівнює їх із векторами, які були отримані на попередніх етапах та збережені в пам'яті системи. Порівняння відбувається на основі обчислення відстані між векторами, що дозволяє визначити ступінь схожості між обличчями;
 7. Виявлення збігу: Після порівняння система вирішує, чи є збіг між векторами. Якщо збіг знайдено, система ідентифікує особу та відображає її ім'я. Якщо ж збіг відсутній, система виводить повідомлення «Не виявлено».
- Додатково, система обробляє зображення в умовах різної якості та освітлення, що впливає на точність розпізнавання. Однак завдяки використанню

векторного представлення облич, система здатна працювати навіть за складних умов. Зображення виступають основним джерелом для формування та збереження шаблонів, а кодування у векторні репрезентації дозволяє зберегти необхідну точність та швидкість обробки при порівнянні з новими обличчями.

2.4 Побудова та налаштування системи

Під час побудови та налаштування системи розпізнавання облич, статі та віку, необхідно дотримуватися певної структури проєкту (рис. 2.4).

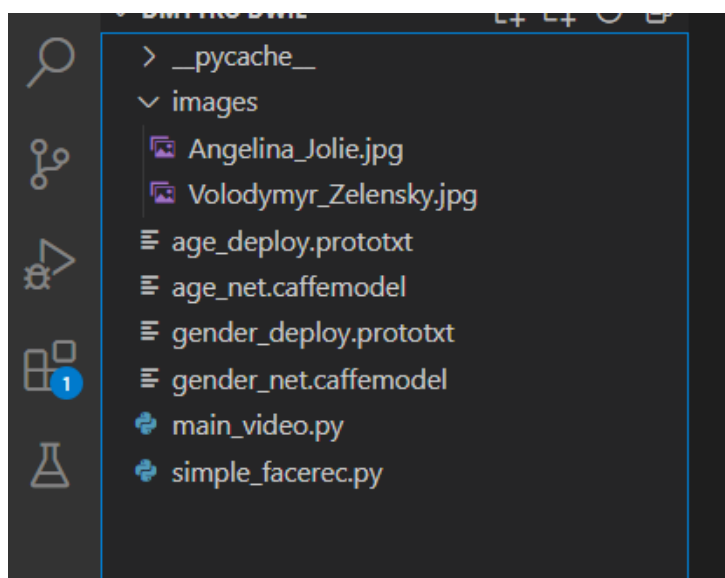


Рисунок 2.4 — Структура проєкту

Структура проєкту для розпізнавання облич у реальному часі містить кілька важливих компонентів, кожен із яких відповідає за певний аспект функціонування системи. Основна частина структури включає файли зображень для ідентифікації облич, які попередньо завантажуються в папку проєкту, а також моделі для визначення віку та статі, що використовуються для аналізу в реальному часі. Окрім зображень, необхідно підготувати та завантажити відповідні моделі для розпізнавання вікових груп і статі, що зберігаються у файлах:

- `age_deploy.prototxt` і `age_net.caffemodel` — для аналізу та визначення віку людини;
- `gender_deploy.prototxt` і `gender_net.caffemodel` — для визначення статі людини.

Ці моделі побудовані на основі архітектури Caffe та забезпечують точність визначення статі й вікових груп на основі нейронних мереж. Щоб система коректно працювала, важливо, щоб ці моделі були завантажені та правильно інтегровані в код проєкту. Окрім моделей, значну роль відіграють бібліотеки, які відповідають за різні етапи обробки даних. Вони дозволяють працювати з зображеннями та відео, розпізнавати обличчя та відображати результати. Основні бібліотеки, які використовуються у проєкті:

- `face_recognition` ключова бібліотека для розпізнавання облич, яка використовує глибокі нейронні мережі для виявлення та кодування облич. Вона відповідає за порівняння зображень користувача з базою даних і виведення результатів;
- `OpenCV (cv2)` надає інструменти для захоплення відео-потоків з камери та обробки кожного кадру. `OpenCV` виконує операції з обробки зображень і працює із відео-потокami в реальному часі;
- `PIL (Python Imaging Library)` використовується для малювання тексту на зображеннях, таких як ім'я, стать та вік користувача, що виводяться поряд із виявленим обличчям;
- `numpy` відповідає за обробку багатовимірних масивів даних, що необхідні для математичних операцій над зображеннями та кодуваннями облич.

Для належного функціонування системи необхідно переконатися, що всі ці бібліотеки встановлені, що можна зробити за допомогою менеджера пакетів `pip`.

Окрім цього, структура проєкту повинна бути добре організована, щоб забезпечити легку інтеграцію нових компонентів і спрощення майбутньої модифікації. Зокрема, необхідно підтримувати чіткий поділ між зображеннями, моделями та основним кодом, що полегшить керування ресурсами та

масштабування проєкту. Загалом, правильна підготовка моделей, налаштування бібліотек і чітка структура проєкту є ключовими факторами для стабільної та ефективної роботи системи розпізнавання обличчя в реальному часі.

2.4.1 Обробка відео-потoku за допомогою OpenCV

Обробка відео-потoku є критичним етапом для системи розпізнавання обличчя в реальному часі. Використання бібліотеки OpenCV дозволяє не лише захоплювати відео з камери, але й обробляти кожен кадр у режимі реального часу, накладати результати аналізу на зображення та виводити їх на екран для користувача. Це забезпечує плавність роботи та оперативне реагування системи на зміни в кадрі.

Для захоплення відео-потoku використовується функція `cv2.VideoCapture(0)`, яка дозволяє отримати доступ до першої підключеної камери. Це є початковим етапом роботи системи, на якому зчитується відео у вигляді послідовності кадрів для подальшого аналізу.

```
python
Копіювати код
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        print("Не вдалося отримати кадр з камери")
        break
```

Аргумент 0 у функції `cv2.VideoCapture(0)` вказує на першу камеру, що підключена до комп'ютера. Система починає захоплення відео-потoku і на кожному кроці зчитує новий кадр за допомогою методу `cap.read()`. Змінна `frame` містить кожен зчитаний кадр, а значення `ret` інформує про те, чи вдалося отримати кадр. Якщо кадр не був отриманий (наприклад, через проблеми з камерою), виводиться повідомлення про помилку і система завершує роботу.

Захоплені кадри використовуються для подальшої обробки в системі розпізнавання облич. Оскільки OpenCV за замовчуванням використовує формат BGR для зображень, але більшість алгоритмів глибокого навчання вимагають

формат RGB, необхідно виконати перетворення кольорів для коректної роботи з алгоритмами розпізнавання облич:

```
python
Копіювати код
rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
```

Функція `cv2.cvtColor()` перетворює кадр з BGR у RGB. Це є необхідним етапом перед передачею кадру до алгоритму розпізнавання облич для забезпечення коректного функціонування. Далі оброблений кадр передається на алгоритми, де відбувається розпізнавання облич, визначення віку та статі користувача.

Щойно обличчя виявлено та проаналізовано, результати, такі як ім'я, стать, вік та рамка навколо обличчя, накладаються на кадр. OpenCV дозволяє зручно виводити оброблений кадр назад на екран у вікні, де користувач може спостерігати за результатами роботи системи в реальному часі:

```
python
Копіювати код
cv2.imshow("Frame", frame)
if cv2.getWindowProperty('Frame', cv2.WND_PROP_VISIBLE) < 1:
    break
```

Функція `cv2.imshow()` відображає оброблений кадр у вікні з назвою "Frame". На цьому етапі користувач може бачити рамки навколо виявлених облич і текстові написи з інформацією про кожне обличчя (ім'я, стать, вік). Система також перевіряє, чи вікно було закрито користувачем. Якщо вікно закрито, система припиняє обробку відео та переходить до завершення роботи.

Коли система закінчує свою роботу (наприклад, при натисканні клавіші ESC або закритті вікна), камера звільняється, і всі використані ресурси очищаються. Це важливий етап для коректного завершення роботи програми:

```
python
Копіювати код
cap.release()
cv2.destroyAllWindows()
```

Метод `cap.release()` закриває з'єднання з камерою, дозволяючи звільнити ресурси, які використовувались для захоплення відео. `cv2.destroyAllWindows()`

закриває всі вікна, відкриті під час роботи системи, і очищає їх із пам'яті. Цей етап важливий для забезпечення стабільної роботи системи та запобігання непередбачуваним помилкам під час повторного запуску або при використанні інших програм.

Завдяки OpenCV система здатна захоплювати, обробляти та відображати відео-потік у реальному часі. Кожен кадр проходить через процес виявлення та розпізнавання облич, а результати накладаються на зображення й виводяться користувачеві у зручному вигляді. Це робить систему корисною для широкого спектра застосувань, від контролю доступу до відеоспостереження або аналізу поведінки користувачів.

2.4.2 Використання моделей для визначення віку та статі з Caffe

У системі для визначення віку та статі облич використовується набір моделей на основі Caffe. Caffe це фреймворк для глибокого навчання, який надає можливість використовувати попередньо натреновані моделі для різних задач, таких як класифікація зображень, розпізнавання об'єктів та облич. У системі використовується дві попередньо натреновані моделі: для визначення віку та статі.

Для завантаження моделей у Caffe використовується функція `cv2.dnn.readNetFromCaffe()`. Ці моделі мають формат `.prototxt` для конфігураційної частини та `.caffemodel` для збереження ваг:

```
gender_net =
cv2.dnn.readNetFromCaffe('gender_deploy.prototxt',
'gender_net.caffemodel')
age_net = cv2.dnn.readNetFromCaffe('age_deploy.prototxt',
'age_net.caffemodel')
```

Функція `cv2.dnn.readNetFromCaffe()` завантажує структуру нейронної мережі із конфігураційного файлу `.prototxt` і ваги з моделі `.caffemodel`.

У системі використано дві окремі моделі для визначення статі (`gender_net`) та віку (`age_net`). Обидві моделі були натреновані на великих наборах даних для забезпечення високої точності прогнозів.

Щоб нейронна мережа могла правильно обробити зображення обличчя для визначення віку та статі, зображення спочатку потрібно перетворити у формат, придатний для обробки мережею. Для цього використовується функція `cv2.dnn.blobFromImage()`, яка підготовлює зображення для подачі в нейронну мережу:

```
blob = cv2.dnn.blobFromImage(face_img, 1.0, (227, 227), (104.0,
177.0, 123.0), swapRB=False)
```

Функція `cv2.dnn.blobFromImage()` створює «blob» вхідний тензор, що використовується для подачі зображення в нейронну мережу. Вона нормалізує пікселі зображення та масштабує його до необхідного розміру, у цьому випадку 227x227 пікселів.

Параметри (104.0, 177.0, 123.0) використовуються для віднімання середніх значень зображення, що є стандартною процедурою при роботі з нейронними мережами для обробки зображень.

Після підготовки зображення воно подається в модель для визначення статі, яка передбачає, чи є людина на зображенні чоловіком або жінкою:

```
gender_net.setInput(blob)
gender_preds = gender_net.forward()
gender = gender_list[gender_preds[0].argmax()]
```

Функція `setInput()` подає підготовлене зображення (blob) в модель для обробки. Функція `forward()` виконує прогноз на основі зображення та повертає масив ймовірностей для кожного класу (чоловік/жінка). Використовується функція `argmax()`, щоб вибрати клас з найвищою ймовірністю, який і буде визначати стать.

Після визначення статі обличчя система переходить до визначення віку, використовуючи аналогічний процес:

```
age_net.setInput(blob)
age_preds = age_net.forward()
age = age_list[age_preds[0].argmax()]
```

Після подачі «blob» в модель для визначення віку, модель прогнозує віковий діапазон на основі зображення. Використовується аналогічний підхід із `argmax()`,

щоб вибрати найбільш імовірний віковий діапазон з попередньо визначеного списку вікових груп, таких як «0-2», «15-20», «38-43» тощо.

Після визначення статі та віку система відображає результати на кадрі за допомогою OpenCV, де додається текст з інформацією про стать та вік людини:

```
gender_text = f"Стать: {gender}"
draw.text((x1 + 5, y1 + 5), gender_text, font=font, fill=(255,
255, 255))
age_text = f"Вік: {age}"
draw.text((x1 + 5, y1 + gender_height + 15), age_text, font=font,
fill=(255, 255, 255))
```

Використовується бібліотека PIL для додавання тексту на кадр. Система відображає стать і вік поруч з обличчям, яке було розпізнано на відео. Використання Caffe для визначення статі та віку забезпечує високу точність і швидкість обробки, що дозволяє отримувати точні прогнози в режимі реального часу. Моделі Caffe були попередньо натреновані на великих наборах даних, що робить їх ідеальними для застосування в таких задачах, як розпізнавання облич, віку та статі у відео-потоці.

2.4.3 Реалізація розпізнавання облич із використанням Face Recognition

Для реалізації розпізнавання облич у системі використовується популярна бібліотека Face Recognition, яка забезпечує ефективні алгоритми для виявлення та розпізнавання облич. Face Recognition базується на алгоритмах глибокого навчання, що дозволяє швидко й точно ідентифікувати обличчя у відео або на зображеннях. Ця бібліотека широко використовується завдяки своїй простоті у використанні та високій точності.

Першим кроком у системі є завантаження всіх зображень облич, які будуть використовуватися для розпізнавання. Ці зображення зберігаються у певній директорії («images»), і кожне зображення асоціюється з конкретною особою. Face Recognition дозволяє просто кодувати обличчя у векторні представлення, які містять унікальні характеристики кожного обличчя:

```
python
```

```
Копіювати код
sfr = SimpleFacerec()
sfr.load_encoding_images("images/")
```

Функція `load_encoding_images()` завантажує всі зображення з теки «images» і перетворює їх у векторні представлення (face encodings). Це дозволяє системі порівнювати нові обличчя з базою даних, що була створена з раніше збережених зображень. Процес кодування є ключовим для ідентифікації, оскільки дозволяє зберегти унікальні риси кожного обличчя, які потім використовуються для порівняння з новими обличчями.

Після завантаження та кодування зображень, система переходить до обробки відео-потoku в реальному часі, який отримується з камери. Для кожного кадру відео використовується функція для виявлення облич. Це дозволяє визначити, де саме на кадрі знаходяться обличчя, та отримати їхні координати для подальшого аналізу:

```
python
Копіювати код
face_locations, face_names = sfr.detect_known_faces(frame)
```

Функція `detect_known_faces()` визначає обличчя у кадрі та повертає дві основні змінні: `face_locations` — координати виявлених облич, та `face_names` — імена відповідних осіб, або "Не виявлено", якщо обличчя не знайдено у базі даних. Це дозволяє системі зрозуміти, які обличчя вже відомі, а які потрібно додатково обробити. Після виявлення облич кожне з них перетворюється на векторне представлення для порівняння з базою даних.

Для цього використовується функція `face_encodings()`, яка генерує векторні представлення для кожного обличчя, виявленого у кадрі:

```
python
Копіювати код
face_encodings =
face_recognition.face_encodings(rgb_small_frame, face_locations)
```

Ці вектори є унікальними для кожної особи та використовуються для порівняння з уже збереженими обличчями. Це ключовий момент у роботі системи, оскільки саме векторизація дозволяє точно ідентифікувати обличчя за його

особливостями. Система потім порівнює отримані векторні представлення з тими, що були збережені на етапі завантаження зображень.

Процес порівняння виконується за допомогою функції `compare_faces()`, яка порівнює вектори знайдених облич із тими, що вже є в базі даних:

```
python
Копіювати код
matches =
face_recognition.compare_faces(self.known_face_encodings,
face_encoding)
name = "Не виявлено"
face_distances =
face_recognition.face_distance(self.known_face_encodings,
face_encoding)
best_match_index = np.argmin(face_distances)
if matches[best_match_index]:
    name = self.known_face_names[best_match_index]
```

Функція `compare_faces()` порівнює отримані вектори зі збереженими, щоб визначити, чи є збіг. Якщо збіг знайдено, система ідентифікує особу і виводить її ім'я. Якщо ж збігу немає, система повертає повідомлення "Не виявлено". Щоб визначити найближчий збіг, використовується функція `face_distance()`, яка розраховує відстань між векторами, і найменша відстань вказує на найімовірніший збіг.

Окрім ідентифікації, система також відображає результати у вигляді кольорових рамок навколо облич. Зелені рамки використовуються для впізнаних облич, а червоні для тих, що не були знайдені у базі даних:

```
python
Копіювати код
if name != "Не виявлено":
    box_color = (23, 163, 85)
else:
    box_color = (221, 51, 51)
draw.rectangle(((x1, y1), (x2, y2))), outline=box_color, width=4)
draw.text((x1 + 5, y1 - text_height - 5), name, font=font,
fill=(255, 255, 255))
```

Зелена рамка та ім'я відображаються для розпізнаних облич, а червона рамка — для невідомих облич. Це дає користувачеві можливість легко

відстежувати процес розпізнавання в реальному часі. Для візуалізації тексту та рамок використовується бібліотека PIL.

Для наочного розуміння роботи системи та взаємодії всіх її компонентів разом, можна використати блок-схему, яка демонструє послідовність виконання алгоритму (рис. 2.5).

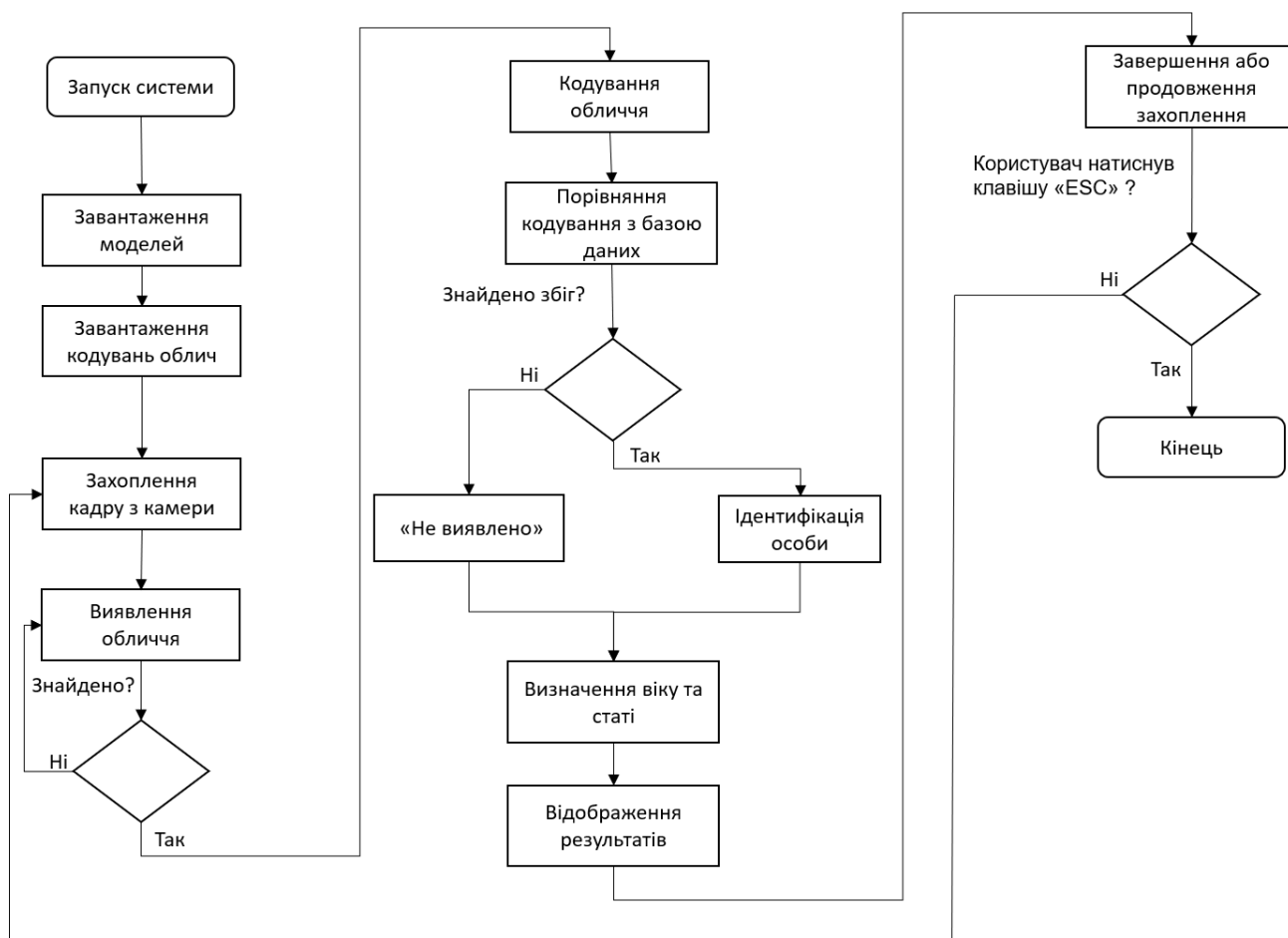


Рисунок 2.5 — Блок-схема роботи системи

Блок-схема роботи системи відображає основні етапи її функціонування, починаючи від запуску і завантаження необхідних моделей до визначення особи, віку та статі. Спершу система ініціюється: завантажуються моделі для розпізнавання обличчя, віку та статі, а також кодування обличчя із папки для подальшого порівняння з вхідними даними. Після цього система починає захоплювати кадри з відео-потoku камери в режимі реального часу. На кожному

кадрі відбувається виявлення облич. Якщо обличчя було знайдено, воно кодується у векторне представлення, яке згодом порівнюється з наявними кодуваннями в базі даних. У разі збігу система ідентифікує особу і переходить до визначення віку та статі за допомогою попередньо завантажених моделей. Якщо збігу не знайдено, система виводить повідомлення «Не виявлено» і так само визначає вік і стать.

Незалежно від результату ідентифікації, система відображає на екрані ім'я особи (або «Не виявлено»), стать та вік, і малює рамку навколо обличчя. Після цього система перевіряє, чи натиснув користувач клавішу «ESC» для завершення роботи. Якщо натиснуто, система закінчує свою роботу, якщо ні процес повторюється знову, починаючи з захоплення наступного кадру.

Система побудована на основі алгоритмів розпізнавання облич, статі та віку в режимі реального часу. Вона автоматично виявляє та обробляє обличчя в кадрі, надаючи користувачу інформацію про ідентифіковану особу, або повідомляючи про відсутність збігів. Завдяки використанню моделей для аналізу віку та статі, система забезпечує додаткову функціональність, що робить її універсальним рішенням для розпізнавання в різних умовах.

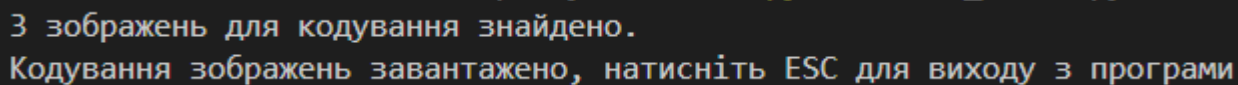
3 ТЕСТУВАННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ

3.1 Тестування системи в реальному часі

Тестування системи розпізнавання обличчя в реальному часі є важливим етапом для оцінки її продуктивності, точності та надійності. Під час тестування можна виявити потенційні проблеми, такі як помилкові спрацьовування, затримки в обробці відео-потoku або помилки в розпізнаванні облич, віку та статі. Це дозволяє оптимізувати систему, перевірити її роботу в різних умовах (освітлення, ракурси) і забезпечити її коректну взаємодію з користувачем. Тестування також допомагає переконатися, що всі модулі та компоненти системи працюють згідно з вимогами.

3.2 Аналіз точності розпізнавання облич, визначення віку та статі

Коли користувач запускає систему розпізнавання облич, на екрані відразу з'являється відео-потік із камери, що працює в реальному часі. Одночасно у консолі відображається повідомлення, яке інформує про кількість зображень, які було знайдено та підготовлено для кодування облич (рис. 3.1).



```
3 зображень для кодування знайдено.  
Кодування зображень завантажено, натисніть ESC для виходу з програми
```

Рисунок 3.1 — Повідомлення в консолі

Це вказує на те, що система готова до роботи, і користувач може спостерігати результати безпосередньо на екрані.

Після захоплення кадру з відео-потoku система починає обробляти обличчя користувача. Для кожного знайденого обличчя система малює рамку навколо нього та визначає, чи є це обличчя вже в базі даних. Якщо обличчя не знайдено в теці

«images», система виводить червону рамку з повідомленням «Не виявлено», що означає, що зображення користувача не було заздалегідь додане до бази (рис. 3.2).



Рисунок 3.2 — Демонстрація не ідентифікованої особи

Якщо ж обличчя було попередньо завантажено в теку «images», система швидко порівнює його із наявними кодуваннями, ідентифікує ім'я користувача та виводить результат у форматі зеленої рамки навколо обличчя. Ім'я користувача відображається в межах цієї рамки (рис. 3.3)

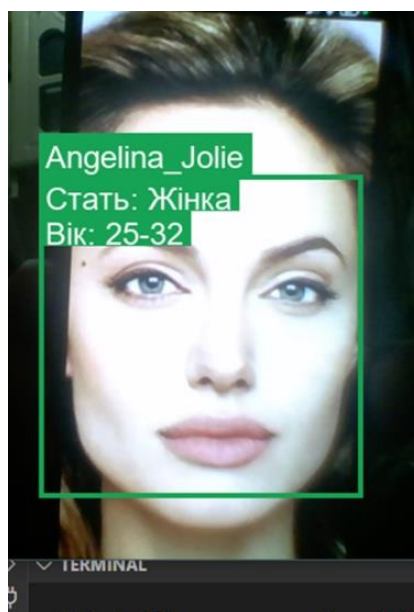


Рисунок 3.3 — Демонстрація ідентифікованої особи

Крім імені, система також виводить додаткову інформацію про стать та вік людини. Стать і вік визначаються за допомогою моделей глибокого навчання, що були попередньо завантажені. Ця інформація також додається до рамки біля обличчя. Таким чином, користувач не тільки бачить своє ім'я (якщо воно є в базі), але й отримує інформацію про свою стать та вік (рис. 3.4).

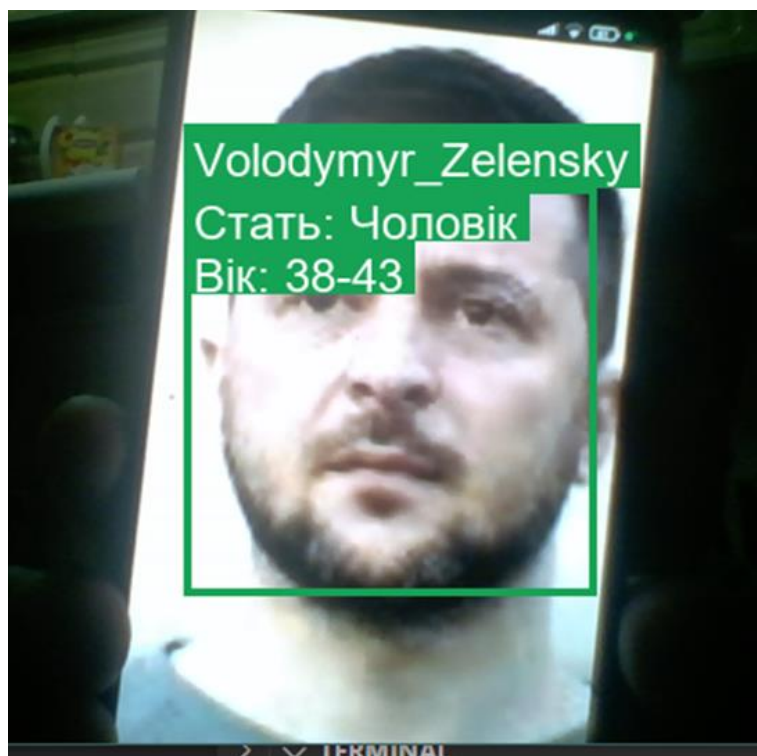


Рисунок 3.4 — Демонстрація не ідентифікованої особи, віку та статті

Важливо зазначити, що через певні технічні обмеження деякі бібліотеки системи не працюють із українською розкладкою клавіатури. Через це ім'я користувача завжди відображається англійською мовою, навіть якщо його запис було зроблено іншими мовами. Це не впливає на інші функціональні аспекти системи, проте може бути враховано при подальшій адаптації програмного забезпечення. Система продовжує працювати в реальному часі, ідентифікуючи всі нові обличчя, які з'являються у відео-потоці. Щоб завершити роботу, користувач може натиснути клавішу «ESC», що закриє програму та зупинить обробку кадрів.

3.3 Тестування продуктивності при різних умовах освітлення

Тестування системи розпізнавання обличчя у різних умовах освітлення показало, що вона демонструє стабільну роботу навіть у складних умовах. Зокрема, система тестувалася в умовах слабого освітлення, де освітлення практично відсутнє, а кадр сповнений тіней (рис. 3.5).

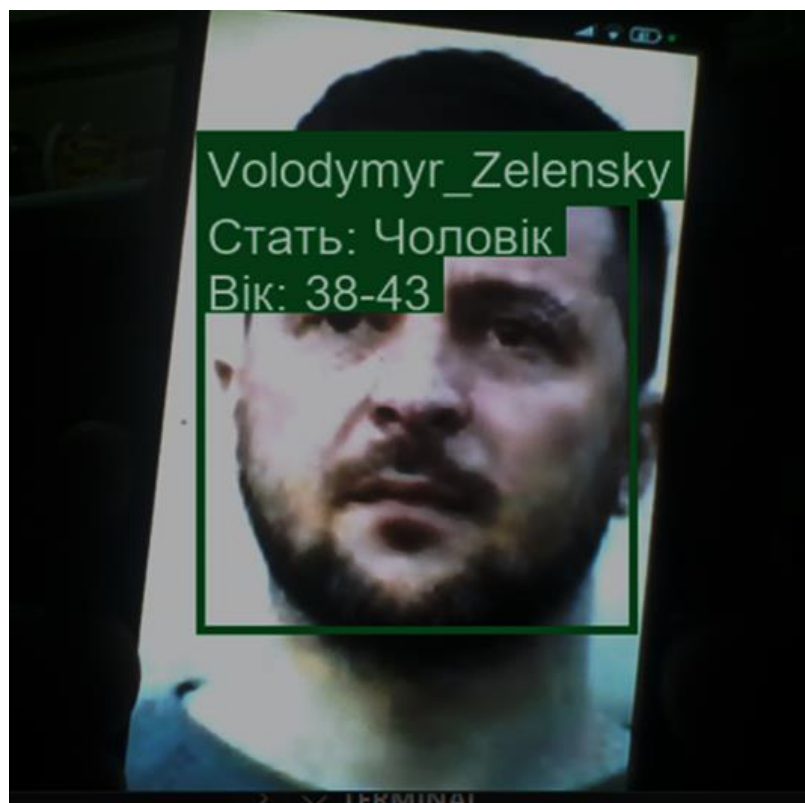


Рисунок 3.5 — Демонстрація роботи системи при поганому освітленні

Незважаючи на такі несприятливі умови, система успішно виявила обличчя, визначила стать і вік користувача та вивела відповідну інформацію. Проте, що важливо зазначити, в умовах низького освітлення процес стабілізації параметрів, таких як вік та стать, займає більше часу, ніж за стандартних умов. Це відбувається через складність виявлення деталей обличчя, які важливі для точного визначення характеристик.

В умовах через мірного освітлення коли на кадр потрапляє надлишкова кількість світла. В таких умовах частина обличчя може бути занадто яскравою, що також ускладнює розпізнавання (рис. 3.6).

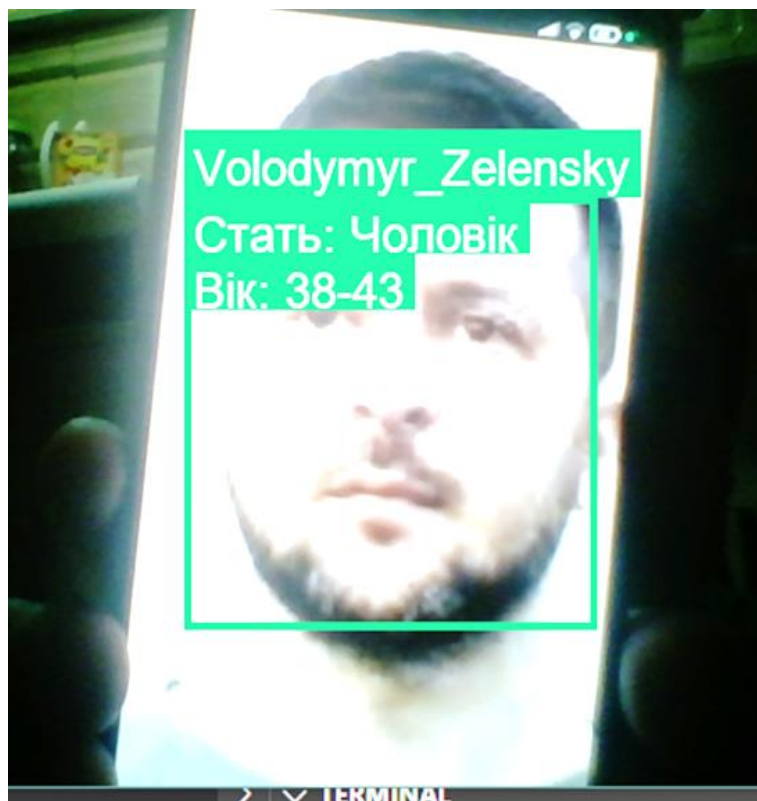


Рисунок 3.6 — Демонстрація роботи системи при через мірному освітленні

Однак, навіть при такому значному освітленні, система здатна правильно виявити обличчя та провести ідентифікацію. Як і в умовах низького освітлення, процес визначення віку та статі вимагає додаткового часу через складність аналізу перенасичених світлом областей обличчя.

В обох випадках система залишається функціональною, але точність і швидкість визначення параметрів можуть варіюватися залежно від умов освітлення. Це свідчить про надійність алгоритму, але також підкреслює необхідність додаткової оптимізації для забезпечення стабільної роботи в умовах різких змін освітлення.

3.4 Використання різних зображень для оцінки моделі

Під час тестування системи розпізнавання обличч важливо перевірити її здатність точно виявляти вік і стать людей різних вікових категорій. Особливої уваги заслуговують фактори, які можуть ускладнювати процес розпізнавання, такі як окуляри, капелюхи або інші перешкоди, що частково закривають обличчя. Також варто оцінити, як система справляється з розпізнаванням обличч із яскраво вираженими емоціями, оскільки це може впливати на точність виявлення віку та статі. Крім того, тестування на зображеннях із кількома людьми допомагає перевірити, чи зможе система одночасно коректно ідентифікувати кожне обличчя.

На рисунку 3.7 видно що система успішно розпізнала обличчя хлопця з точним визначенням віку.

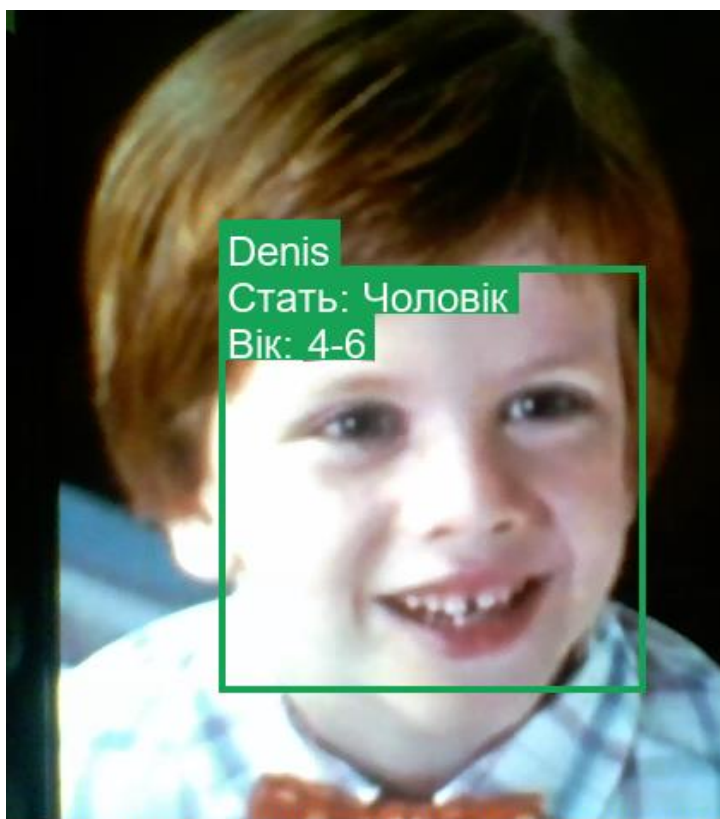


Рисунок 3.7 — Демонстрація роботи системи для малої вікової групи

Хоча процес виявлення тривав трохи довше через необхідність додаткової обробки даних, система коректно вивела значення імені, віку та статі.

На рисунку 3.8 показано розпізнавання дівчини.

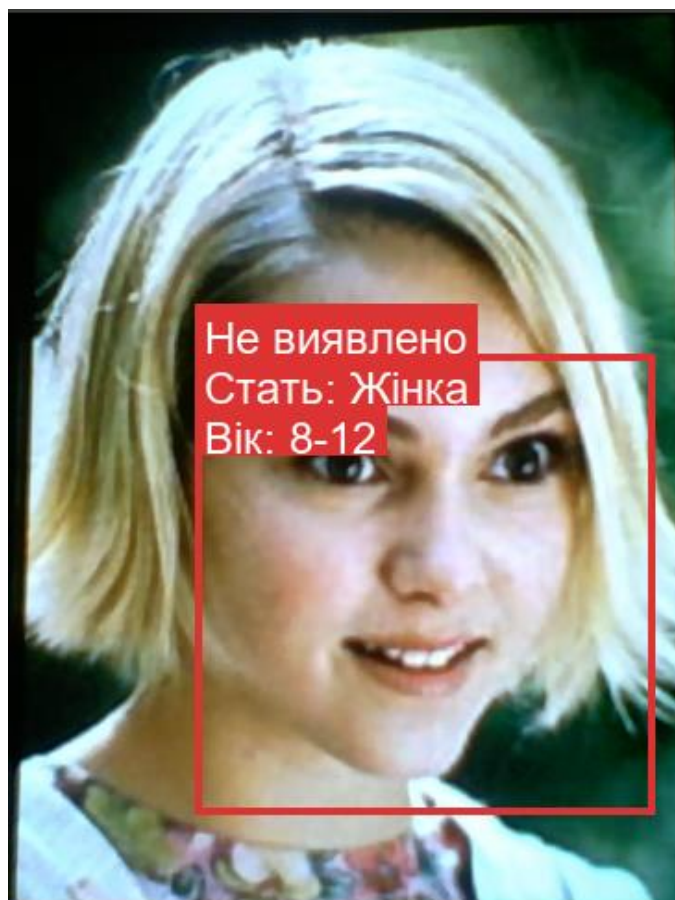


Рисунок 3.8 — Демонстрація роботи системи для малої вікової групи

Обличчя дівчини не було завантажено в базу даних, тому система вивела повідомлення «Не виявлено», але водночас правильно визначила стать і приблизний вік. Це демонструє, що навіть при відсутності збігу в базі даних, система здатна надійно визначати основні параметри.

Таким чином, система коректно виявляє вік та стать для цієї вікової категорії, хоча для точного визначення віку, як і очікувалося, потребується трохи більше часу через специфіку алгоритму.

Тестування системи для старшої категорії осіб показує, що система успішно розпізнає обличчя і коректно визначає параметри віку та статі. У першому випадку (рис. 3.9) система розпізнала відоме обличчя, вказавши його ім'я, стать та приблизний вік, відображаючи дані в зеленій рамці. Це підтверджує ефективність системи для визначення параметрів навіть у випадку зі старшою категорією осіб.

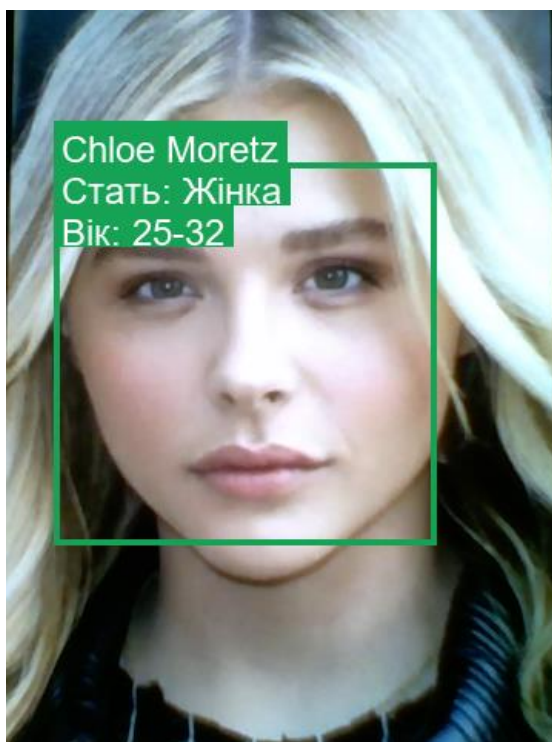


Рисунок 3.9 — Демонстрація роботи системи для середньої вікової групи

У другому випадку, обличчя не було попередньо завантажено в базу даних, тому система вивела червону рамку та повідомлення «Не виявлено» (рис. 310).

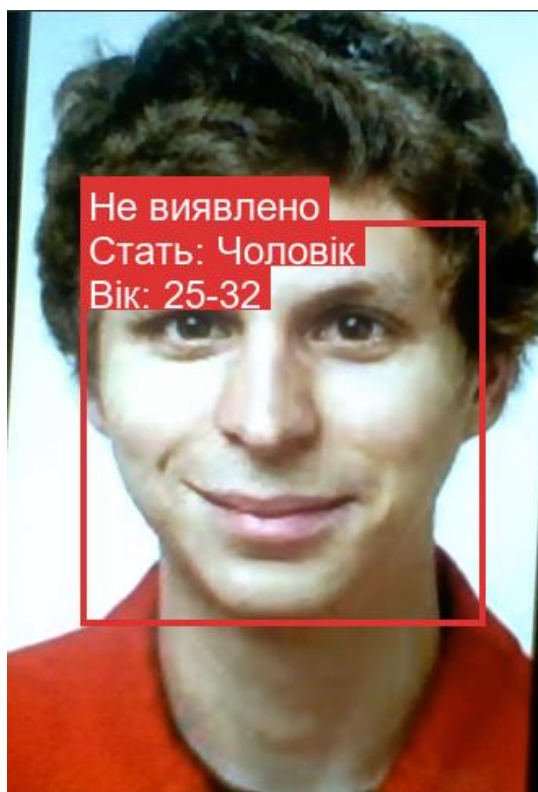


Рисунок 3.10 — Демонстрація роботи системи для середньої вікової групи

Однак, навіть у випадку невідомих облич, система коректно визначає стать і приблизний вік, що демонструє її стабільну роботу з різними групами людей.

Тестування системи на літній віковій групі людей продемонструвало, що система здатна точно визначати як вік, так і стать. На першому зображенні система успішно розпізнала відоме обличчя, ідентифікувала його стать та вік, що відповідає реальності на момент створення фото (рис. 3.11). Рамка зелена, що означає, що обличчя було знайдено у базі даних.

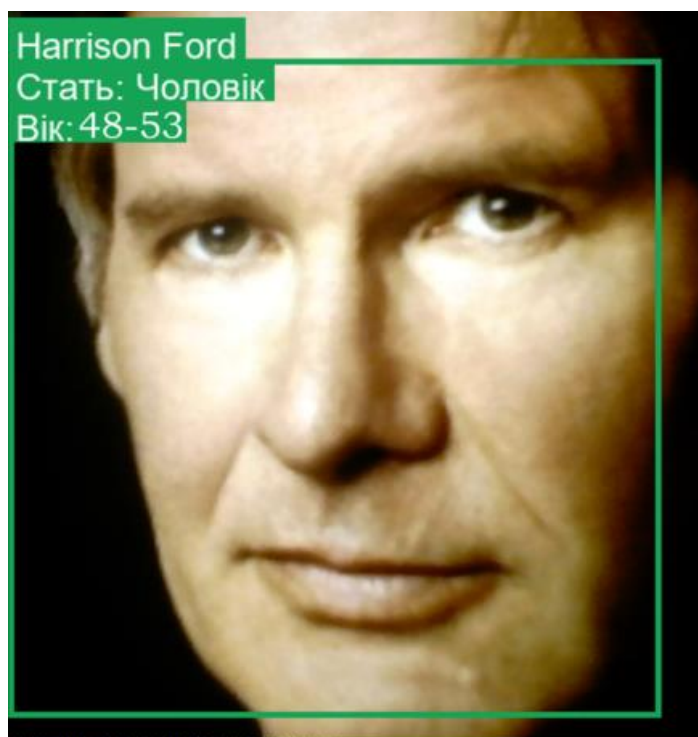


Рисунок 3.11 — Демонстрація роботи системи для літньої вікової групи

На другому зображенні система розпізнала обличчя літньої жінки, але воно не було знайдено у базі даних, тому система відобразила червону рамку з повідомленням «Не виявлено». Незважаючи на відсутність збігу у базі, система все одно правильно визначила стать та вік, позначивши їх (рис. 3.12).

У процесі тестування системи на цьому прикладі виникли певні сумніви щодо точності визначення віку. Жінка на зображенні здається старшою за вік, визначений системою (48-53 роки). Це може бути пов'язано з сильно освітленим

фото, через що алгоритм не може захопити всі необхідні риси обличчя для точного аналізу.

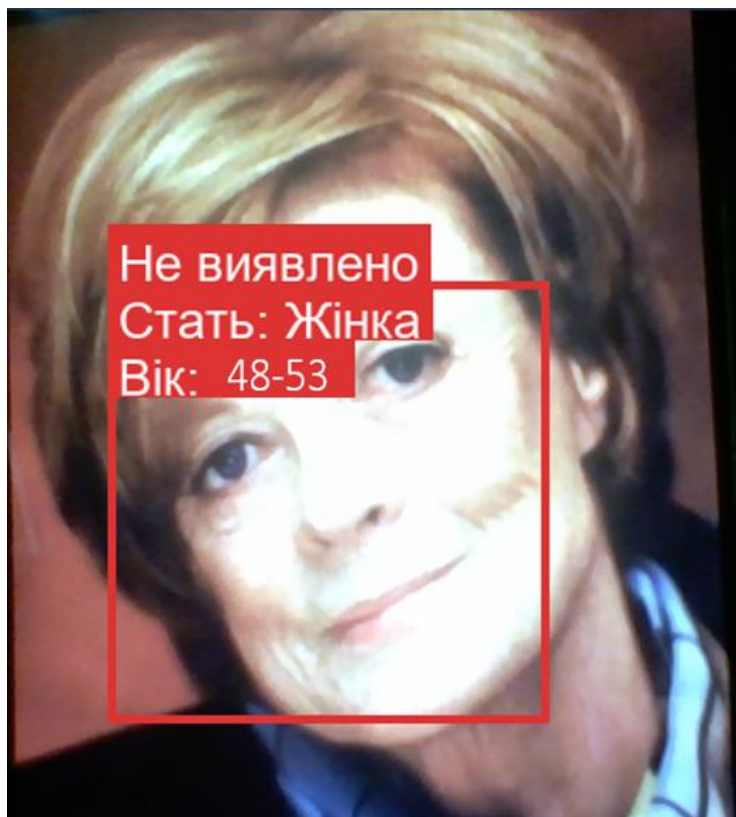


Рисунок 3.12 — Демонстрація роботи системи для літньої вікової групи

Ці результати підтверджують, що система може коректно працювати з різними віковими категоріями, включно зі літньою віковою групою, надаючи результати навіть у випадках, коли особа не знайдена в базі та має якісь дефекти у фото.

Система розпізнавання облич демонструє високу ефективність у різних умовах, зокрема при частковому закритті обличчя аксесуарами, такими як окуляри або капелюх. З лівої сторони на зображенні представлено завантажене фото Леонардо Ді Капріо, яке система використовує для створення векторного кодування обличчя, що служить для подальшого порівняння з іншими зображеннями (рис. 3.13).

На правій стороні зображення показано, як система успішно ідентифікує того ж самого актора, незважаючи на те, що частина його обличчя закрита сонцезахисними окулярами та шляпою (рис. 3.13).



Рисунок 3.13 — Завантажене фото в систему, та демонстрація виявлення обличчя системою

Цей тест показує, що алгоритм має достатню стійкість до перешкод, які можуть виникати через часткове закриття обличчя. Окрім того, система коректно визначає стать та вік, що додає додатковий рівень точності в розпізнаванні. Навіть в умовах, коли деякі ключові риси обличчя (наприклад, очі) не повністю видимі, алгоритм базується на інших доступних рисах для коректної ідентифікації. Це підтверджує надійність системи при роботі з реальними сценаріями, де люди можуть носити окуляри, маски чи інші аксесуари. Таким чином, система є універсальною та готовою для застосування в реальних умовах, де повне відображення обличчя не завжди можливе.

Система розпізнавання обличчя виявляє високу ефективність навіть тоді, коли користувач виражає різні емоції. Незалежно від того, чи це усмішка, сміх, чи серйозний вираз обличчя, алгоритми можуть коректно виявляти та ідентифікувати особу. Це підтверджує надійність системи у сценаріях, де вирази обличчя швидко змінюються, що робить її універсальною та застосовною у різних контекстах використання. На рисунку 3.14 наведено фотографію Джима Керрі, яку завантажено в систему для створення кодування обличчя.

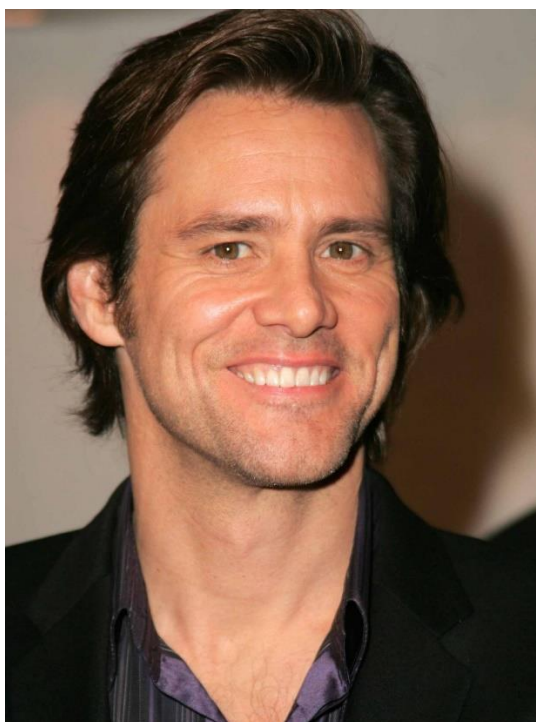


Рисунок 3.14 — Завантажене фото Джима Керрі в систему

На основі наведених прикладів (рис. 1.15) видно, що навіть значні зміни у виразах обличчя не впливають на здатність системи точно ідентифікувати особу. Система вміє адаптуватися до різних емоційних станів, зберігаючи високу точність розпізнавання. Незважаючи на зміни міміки, такі як сміх, серйозність чи інші вирази обличчя, система стабільно і точно відображає ім'я, стать та вік користувача. Це підтверджує, що система адаптивна і може працювати з динамічними змінами виразу обличчя без втрати точності.

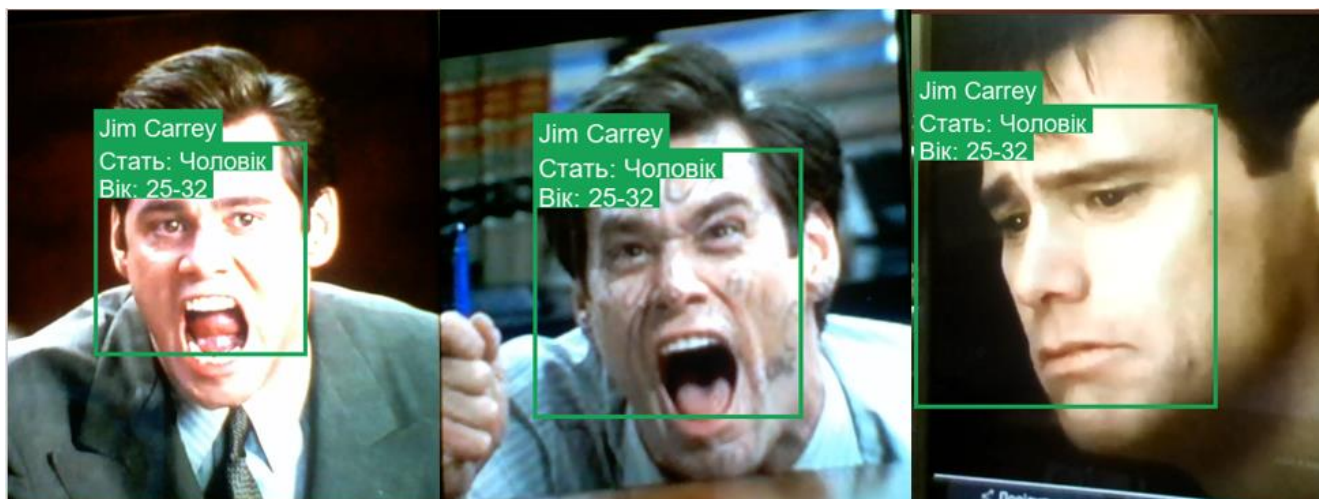


Рисунок 3.15 — Демонстрація роботи системи при різних емоційних станах на обличчі

Ця здатність системи розпізнавати обличчя навіть у разі різних емоцій робить її особливо корисною для застосування в реальних умовах, де емоційні стани користувачів можуть швидко змінюватися.

Система також ефективно працює з парними зображеннями, де на одному кадрі присутні декілька осіб. Як показано на прикладі, система змогла одночасно ідентифікувати двох осіб: Олену Зеленську та Володимира Зеленського (рис. 3.16).

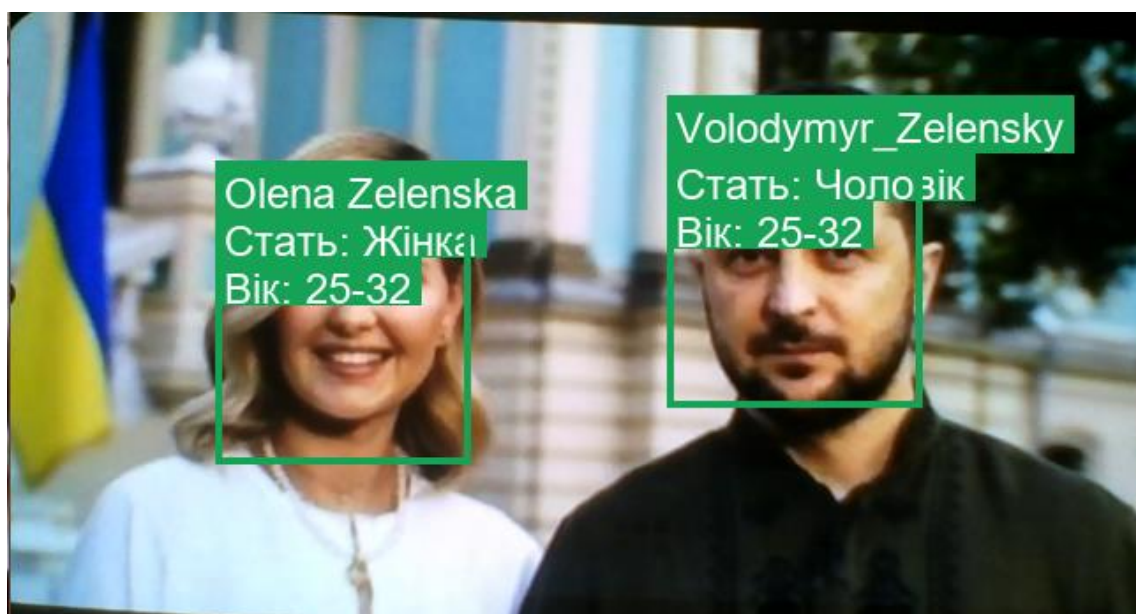


Рисунок 3.16 — Демонстрація роботи системи у випадку коли обличчя на фото 2

Навіть за наявності кількох облич на одному зображенні, алгоритм точно розпізнає їх, визначив статі і вік кожної особи, та правильно вивів інформацію в зелених рамках. Це демонструє, що система добре справляється з розпізнаванням декількох облич одночасно, що є важливим для використання в реальних сценаріях, коли в кадрі можуть бути присутні різні люди.

Коли система обробляє кадри з декількома людьми, вона також успішно розпізнає обличчя, як видно на прикладі із трьома чоловіками (рис. 3.17)

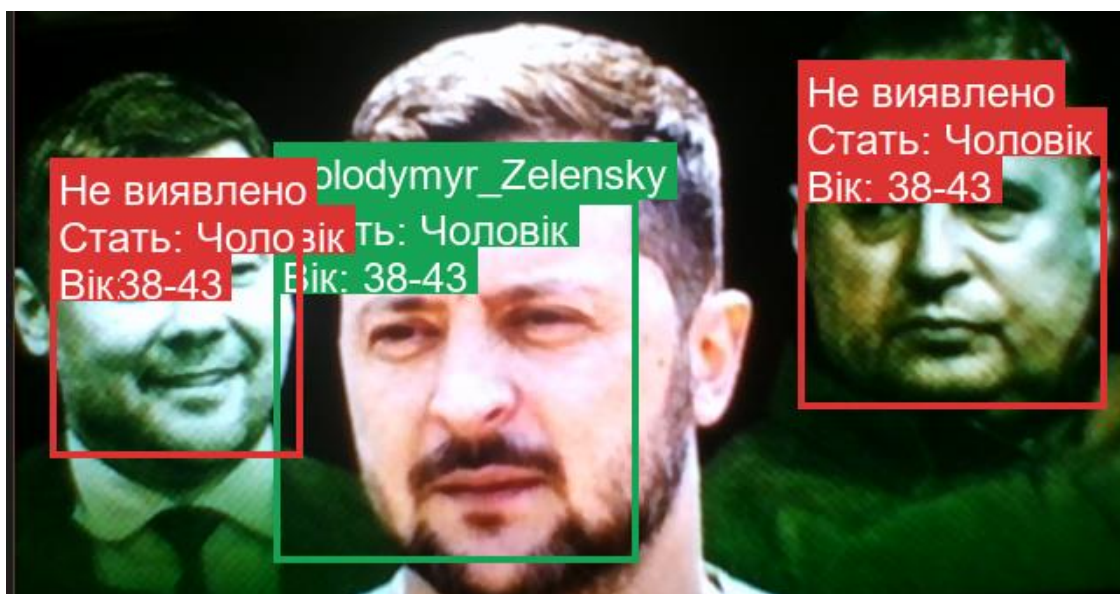


Рисунок 3.17 — Демонстрація роботи системи коли на фото є декілька людей

Система виявила двох людей, яких не було в базі даних, і вивела їх у червоних рамках з відповідним написом «Не виявлено», а для Володимира Зеленського зелену рамку, що означає його успішну ідентифікацію.

Проте, однією з помітних проблем є накладання рамок одна на одну, особливо коли люди перебувають близько один до одного. Це може призвести до часткової перекриття текстової інформації та самої рамки, що знижує зручність сприйняття результатів розпізнавання. У таких випадках можна вдосконалити алгоритм позиціонування рамок або зменшити їх розмір, щоб уникнути накладання.

ВИСНОВКИ

В ході роботи було досліджено методи розпізнавання облич за допомогою штучного інтелекту, а також розроблено комп'ютерну систему, яка здатна виконувати розпізнавання обличчя в реальному часі. Важливість даної теми пояснюється зростаючою потребою в системах автоматизованого розпізнавання облич для забезпечення безпеки та зручності доступу до різних сервісів. Технології штучного інтелекту дозволяють значно підвищити ефективність таких систем, забезпечуючи високу точність розпізнавання та можливість роботи в реальному часі.

Під час розробки системи було використано бібліотеки OpenCV та Face Recognition, а також моделі для визначення віку і статі на базі Caffe. Для тестування було застосовано широкий спектр зображень з різними умовами освітлення, емоціями, та частковим закриттям обличчя (наприклад, окулярами чи головними уборами). Отримані результати показали, що система здатна з високою точністю розпізнавати обличчя навіть в умовах, які ускладнюють процес виявлення.

Система стабільно працювала в режимі реального часу, демонструючи здатність розпізнавати не лише обличчя, але й визначати вік та стать користувачів. У процесі тестування було встановлено, що система може адаптуватися до динамічних змін у зовнішньому вигляді користувачів (зміни виразів обличчя, емоції тощо), а також справлятися з перешкодами, такими як часткове закриття обличчя. Таким чином, розроблена система є ефективним інструментом для розпізнавання облич у реальному часі, що може бути застосована у різних сферах, включаючи безпеку, контроль доступу та інші.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Який принцип роботи систем розпізнавання облич? [Електронний ресурс] // kristall-systems. – 2024. – Режим доступу до ресурсу: https://kristall-systems.net.ua/ua/novosti/kak_rabotaet_raspoznavanie_lits_perspektivyi_tehnologii_v_ukraine/. (дата звернення: 13.08.2022).
2. Medapati, P. K., Tejo Murthy: For IoT-based face recognition system to manage the safety factor in smart cities [Електронний ресурс] // Transactions on Emerging Telecommunications Technologies. – 2020. – Vol. 31, №12. – Режим доступу до ресурсу: researchgate.net/publication/337834311_LAMSTAR_For_IoT-based_face_recognition_system_to_manage_the_safety_factor_in_smart_cities. (дата звернення: 13.09.2024).
3. How do CCTV cameras recognise faces? [Електронний ресурс] // maxnet.ua. – 2024. – Режим доступу до ресурсу: <https://maxnet.ua/en/blog/yak-kameri-videosposterezheniya-rozpiznayut-oblichchya/>. (дата звернення: 13.08.2022).
4. Histogram of Oriented Gradients [Електронний ресурс] // builtin. – 2024. – Режим доступу до ресурсу: <https://builtin.com/articles/histogram-of-oriented-gradients>. (дата звернення: 13.08.2022).
5. Widi Santoso, Rahayu Safitri, Samidi Samidi. Integration of Artificial Intelligence in Facial Recognition Systems for Software Security [Електронний ресурс] // SinkrOn. – 2024. – Т. 8, №2. – С. 1208-1214. – Universitas Budi Luhur. – Режим доступу до ресурсу: <https://www.researchgate.net/publication/380406799> (дата звернення: 13.09.2024).
6. Face Id: Deep learning for face recognition [Електронний ресурс] // Medium. – 2019. – Режим доступу до ресурсу: <https://medium.com/@fenjiro/face-id-deep-learning-for-face-recognition-324b50d916d1>. (дата звернення: 13.09.2024).
7. Using Deep Learning to Design Real-time Face Detection and Recognition Systems [Електронний ресурс] // turing. – 2024. – Режим доступу до ресурсу:

<https://www.turing.com/kb/using-deep-learning-to-design-face-detection-and-recognition-systems>. (дата звернення: 13.09.2024).

8. Zanin M., Atiya N., Basílio J., Baumach J. An Early Stage Researcher's Primer on Systems Medicine Terminology [Електронний ресурс] // Network and Systems Medicine. – 2020. – Т. 4, №1. – С. 2-50. – Режим доступу до ресурсу: https://www.researchgate.net/publication/345903596_An_Early_Stage_Researcher's_Primer_on_Systems_Medicine_Terminology (дата звернення: 13.09.2024).

9. OpenCV Face Recognition [Електронний ресурс] // pyimagesearch. – 2022. – Режим доступу до ресурсу: <https://pyimagesearch.com/2018/09/24/opencv-face-recognition/>. (дата звернення: 13.09.2024).

10. OpenCV Face detection with Haar cascades [Електронний ресурс] // pyimagesearch. – 2021. – Режим доступу до ресурсу: <https://pyimagesearch.com/2021/04/05/opencv-face-detection-with-haar-cascades/>. (дата звернення: 13.09.2024).

11. Meena K., Suruliandi A. Local binary patterns and its variants for face recognition [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://www.researchgate.net/publication/252025071> (дата звернення: 13.09.2024).

12. Singh S., Singh D., Yadav V. Face Recognition Using HOG Feature Extraction and SVM Classifier [Електронний ресурс] // Department of Technical Education, Uttar Pradesh. – 2020. – Т. 8, №9. – С. 6437-6440. – Режим доступу до ресурсу: <https://www.researchgate.net/publication/345903596> (дата звернення: 13.09.2024).

13. A Step-by-Step Guide to Face Detection with the dlib Library [Електронний ресурс] // Medium. – 2023. – Режим доступу до ресурсу: <https://medium.com/@sukanyasingh303/a-step-by-step-guide-to-face-detection-with-the-dlib-library-2e8f6429e632>. (дата звернення: 13.09.2024).

14. Facial Landmark Detection [Електронний ресурс] // Medium. – 2024. – Режим доступу до ресурсу: <https://medium.com/@RiwajNeupane/facial-landmark-detection-a6b3e29eac5b>. (дата звернення: 13.09.2024).

15. Face Recognition [Электронный ресурс] // Github. – 2022. – Режим доступа до ресурсу: https://github.com/ageitgey/face_recognition. (дата звернення: 13.09.2024).
16. tensorflow [Электронный ресурс] // Google. – 2024. – Режим доступа до ресурсу: <https://www.tensorflow.org/>. (дата звернення: 13.09.2024).
17. TensorBoard [Электронный ресурс] // Google. – 2024. – Режим доступа до ресурсу: <https://www.tensorflow.org/tensorboard> (дата звернення: 13.09.2024).
18. Keras [Электронный ресурс] // Keras. – 2024. – Режим доступа до ресурсу: <https://keras.io/>. (дата звернення: 13.09.2024).
19. pytorch [Электронный ресурс] // Facebook AI Research. – 2024. – Режим доступа до ресурсу: <https://pytorch.org/>. (дата звернення: 13.09.2024).
20. MXNet [Электронный ресурс] // Apache Software Foundation. – 2024. – Режим доступа до ресурсу: <https://mxnet.apache.org/versions/1.9.1/>. (дата звернення: 13.09.2024).
21. FaceNet [Электронный ресурс] // geeksforgeeks. – 2022. – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/facenet-using-facial-recognition-system/>. (дата звернення: 13.09.2024).
22. DeepFace: A Popular Open Source Facial Recognition Library [Электронный ресурс] // viso. – 2023. – Режим доступа до ресурсу: <https://viso.ai/computer-vision/deepface/>. (дата звернення: 13.09.2024).
23. Microsoft Face API [Электронный ресурс] // Medium. – 2024. – Режим доступа до ресурсу: <https://medium.com/codenx/azure-face-api-face-detection-recognition-b2b954e3e939>. (дата звернення: 13.09.2024).
24. Analyze Images with the Cloud Vision API [Электронный ресурс] // google. – 2024. – Режим доступа до ресурсу: https://www.cloudskillsboost.google/course_templates/633. (дата звернення: 13.09.2024).
25. Pros and cons of facial recognition [Электронный ресурс] // techtarget.. – 2024. – Режим доступа до ресурсу: <https://www.techtarget.com/whatis/feature/Pros-and-cons-of-facial-recognition>. (дата звернення: 13.09.2024).

26. Performance Analysis of GPU-CPU for The Face Detection [Електронний ресурс] // (EIconCIT) – 2021. – Режим доступу до ресурсу: https://repository.umi.ac.id/655/1/Performance_Analysis_of_GPU-CPU_forThe_Face.pdf. (дата звернення: 13.09.2024).

27. Anwarul S., Dahiya S. A Comprehensive Review on Face Recognition Methods and Factors Affecting Facial Recognition Accuracy [Електронний ресурс] // Graphic Era Hill University. – 2020. – С. 495-514. – Режим доступу до ресурсу: <https://www.researchgate.net/publication/337446642> (дата звернення: 13.09.2024).

28. Розпізнавання обличчя: що потрібно знати про цю технологію [Електронний ресурс] // worldvision. – 2024. – Режим доступу до ресурсу: <https://worldvision.com.ua/articles/raspoznawanie-litsa-hto-nuzhno-znat-ob-etoj-tehnologii>. (дата звернення: 13.09.2024).

29. Ethical compliance for facial recognition technology [Електронний ресурс] // int-comp. – 2024. – Режим доступу до ресурсу: <https://www.int-comp.org/insight/ethical-compliance-for-facial-recognition-technology/>. (дата звернення: 13.09.2024).

30. ЗАКОН УКРАЇНИ Про захист персональних даних [Електронний ресурс] // zakon.rada.gov. – 2024. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>. (дата звернення: 13.09.2024).

31. Caffe (age-and-gender-classification)[Електронний ресурс] // Github. – 2024. – Режим доступу до ресурсу: <https://github.com/eveningglow/age-and-gender-classification/tree/master/model..> (дата звернення: 13.09.2024).

ДОДАТОК А

Лістинг системи

Файл main_video.py:

```
import cv2
from simple_facerec import SimpleFacerec
from PIL import Image, ImageDraw, ImageFont
import numpy as np

# Завантажуємо моделі для визначення статі та віку
gender_net = cv2.dnn.readNetFromCaffe('gender_deploy.prototxt',
'gender_net.caffemodel')
age_net = cv2.dnn.readNetFromCaffe('age_deploy.prototxt',
'age_net.caffemodel')

# Списки для статі та віку
gender_list = ['Чоловік', 'Жінка']
age_list = ['0-2', '4-6', '8-12', '15-20', '25-32', '38-43',
'48-53', '60+']

# Завантаження облич із папки для розпізнавання
sfr = SimpleFacerec()
sfr.load_encoding_images("images/")

# Завантажуємо шрифт для кирилиці
font_path = "arial.ttf"
font = ImageFont.truetype(font_path, 24)

# Відкриваємо камеру
cap = cv2.VideoCapture(0)

while True:
    # Захоплюємо кадр з камери
    ret, frame = cap.read()

    if not ret:
        print("Не вдалося отримати кадр з камери")
        break

    # Розпізнаємо обличчя у кадрі
    face_locations, face_names = sfr.detect_known_faces(frame)

    # Конвертуємо кадр у формат PIL для роботи з кирилицею
    pil_img = Image.fromarray(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB))
    draw = ImageDraw.Draw(pil_img)

    # Обробляємо кожне обличчя у кадрі
    for face_loc, name in zip(face_locations, face_names):
        y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2],
face_loc[3]
```

```

# Вирізаємо обличчя для визначення статі та віку
face_img = frame[y1:y2, x1:x2]
blob = cv2.dnn.blobFromImage(face_img, 1.0, (227, 227),
(104.0, 177.0, 123.0), swapRB=False)

# Визначаємо стать
gender_net.setInput(blob)
gender_preds = gender_net.forward()
gender = gender_list[gender_preds[0].argmax()]

# Визначаємо вік
age_net.setInput(blob)
age_preds = age_net.forward()
age = age_list[age_preds[0].argmax()]

# Визначаємо колір рамки (зелений для впізнаних облич,
червоний для невпізнаних)
if name != "Невиявлено":
    box_color = (23, 163, 85) # Зелений
else:
    box_color = (221, 51, 51) # Червоний

# Малюємо рамку та текст для імені
text_bbox = draw.textbbox((0, 0), name, font=font)
text_width = text_bbox[2] - text_bbox[0]
text_height = text_bbox[3] - text_bbox[1]
draw.rectangle((x1, y1 - text_height - 10), (x1 +
text_width + 10, y1)), fill=box_color)
draw.text((x1 + 5, y1 - text_height - 5), name,
font=font, fill=(255, 255, 255))

# Малюємо текст для статі
gender_text = f"Стать: {gender}"
gender_bbox = draw.textbbox((0, 0), gender_text,
font=font)
gender_width = gender_bbox[2] - gender_bbox[0]
gender_height = gender_bbox[3] - gender_bbox[1]
draw.rectangle((x1, y1), (x1 + gender_width + 10, y1 +
gender_height + 10)), fill=box_color)
draw.text((x1 + 5, y1 + 5), gender_text, font=font,
fill=(255, 255, 255))

# Малюємо текст для віку
age_text = f"Вік: {age}"
age_bbox = draw.textbbox((0, 0), age_text, font=font)
age_width = age_bbox[2] - age_bbox[0]
age_height = age_bbox[3] - age_bbox[1]
draw.rectangle((x1, y1 + gender_height + 10), (x1 +
age_width + 10, y1 + gender_height + age_height + 20)), fill=box_color)
draw.text((x1 + 5, y1 + gender_height + 15), age_text,
font=font, fill=(255, 255, 255))

# Малюємо рамку навколо обличчя

```

```

width=4)        draw.rectangle(((x1, y1), (x2, y2)), outline=box_color,

# Перетворюємо зображення назад у формат OpenCV
frame = cv2.cvtColor(np.array(pil_img), cv2.COLOR_RGB2BGR)

# Відображаємо результат
cv2.imshow("Frame", frame)

# Перевірка, чи вікно закрито
if cv2.getWindowProperty('Frame', cv2.WND_PROP_VISIBLE) <
1:
    break

# Натисніть ESC для виходу
key = cv2.waitKey(1)
if key == 27:
    break

# Звільняємо ресурси
cap.release()
cv2.destroyAllWindows()

```

Файл simple_facerec.py:

```

import face_recognition
import cv2
import os
import glob
import numpy as np

class SimpleFacerec:
    def __init__(self):
        self.known_face_encodings = [] # Список для збереження
кодувань облич
        self.known_face_names = [] # Список для збереження імен

        # Зменшення розміру кадру для пришвидшення обробки
        self.frame_resizing = 0.25

    def load_encoding_images(self, images_path):
        """
        Завантаження зображень для розпізнавання з папки
        :param images_path: шлях до папки з зображеннями
        """
        # Отримати всі файли зображень у папці
        images_path = glob.glob(os.path.join(images_path,
        "*.jpg"))

        print(f"{len(images_path)} зображень для кодування
знайдено.") # Вивести кількість знайдених зображень

```

```

# Для кожного зображення:
for img_path in images_path:
    img = cv2.imread(img_path) # Завантажити зображення
    rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) #
Конвертувати зображення з BGR у RGB

    # Отримати ім'я файлу без розширення
    basename = os.path.basename(img_path)
    (filename, ext) = os.path.splitext(basename)

    # Отримати кодування обличчя з зображення
    img_encoding =
face_recognition.face_encodings(rgb_img)[0]

    # Додати кодування і ім'я до списків
    self.known_face_encodings.append(img_encoding)
    self.known_face_names.append(filename)

    print("Кодування зображень завантажено, натисніть ESC
для виходу з програми")

def detect_known_faces(self, frame):
    # Зменшити розмір кадру для пришвидшення обробки
    small_frame = cv2.resize(frame, (0, 0),
fx=self.frame_resizing, fy=self.frame_resizing)

    # Конвертувати кадр з BGR у RGB для розпізнавання облич
    rgb_small_frame = cv2.cvtColor(small_frame,
cv2.COLOR_BGR2RGB)

    # Знайти обличчя у кадрі
    face_locations =
face_recognition.face_locations(rgb_small_frame)

    # Отримати кодування для знайдених облич
    face_encodings =
face_recognition.face_encodings(rgb_small_frame, face_locations)

    face_names = []
    for face_encoding in face_encodings:
        # Перевірити, чи обличчя збігається з відомими
обличчями
        matches =
face_recognition.compare_faces(self.known_face_encodings,
face_encoding)
        name = "Невиявлено" # Якщо немає збігів, присвоїти
ім'я "Невиявлено"

        # Розрахувати відстань до всіх відомих облич
        face_distances =
face_recognition.face_distance(self.known_face_encodings,
face_encoding)

```

```

        best_match_index = np.argmin(face_distances) #
Знайти індекс найближчого збігу

        # Якщо знайдено збіг, використовувати ім'я
відповідного обличчя
        if matches[best_match_index]:
            name = self.known_face_names[best_match_index]

        # Додати ім'я до списку
        face_names.append(name)

        # Перетворити координати облич з зменшеного розміру
кадру до оригінального
        face_locations = np.array(face_locations) /
self.frame_resizing
        return face_locations.astype(int), face_names #
Повернути координати облич і відповідні імена

```