

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ДЛЯ РОЗПІЗНАВАННЯ ЛЮДСЬКИХ ОБЛИЧ

SOFTWARE FOR HUMAN

FACE RECOGNITION

Виконав(ла): студент(ка) 4 курсу, групи

КНТ-131

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ТРЕТЬЯКОВ Б.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ІЛЬЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ТРЕТЬЯКОВА Богдана Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для розпізнавання людських облич. Software for Human Face Recognition

керівник проєкту (роботи) к.т.н., доцент, ІЛ'ЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “28” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ІЛІЯШЕНКО М.Б., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 28 ” квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Богдан ТРЕТЬЯКОВ
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Матвій ІЛІЯШЕНКО
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
83 с., 3 табл., 26 рис., 3 дод., 18 джерел.

PYCHARM, PYTHON, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ, РОЗПІЗНАВАННЯ ОБЛИЧ.

Об'єкт дослідження – процес розробки програмного забезпечення для розпізнавання образів.

Предмет дослідження – програмні засоби для підтримки процесу розпізнавання людських облич.

Мета роботи – розробка програмного забезпечення для розпізнавання людських облич.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для розпізнавання людських облич. Створено програмне забезпечення для розпізнавання людських облич. Проведено тестування розробленої програмної системи для розпізнавання людських облич.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для розпізнавання людських облич за допомогою мови програмування Python та середовища розробки PyCharm.

Галузь використання – розпізнавання людських облич.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 83 pages, 3 tables, 26 figures, 3 appendixes, 18 sources.

PYCHARM, PYTHON, PROGRAMMING LANGUAGE, SOFTWARE, FACE RECOGNITION.

The object of research is the process of developing software for image recognition.

The subject of the research is software for supporting the process of recognizing human faces.

The purpose of this work is to develop the software for recognizing human faces.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. Project solutions for the creation of software for recognizing human faces has been designed. The software for recognizing human faces has been developed. The developed software system for recognizing human faces has been tested.

Conclusions. The goal of the work has been achieved. The software tools for recognizing human faces using the Python programming language and the PyCharm development environment have been developed.

Scope of use – human face recognition.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для розпізнавання образів	11
1.2 Аналіз програмного забезпечення для підтримки процесу розпізнавання людських облич	21
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання людських облич	36
2.3 Висновки за розділом 2	38
3 Опис програми	40
3.1 Структура програмного забезпечення для розпізнавання людських облич	40
3.2 Функціонування програмного забезпечення для розпізнавання людських облич	41
3.3 Розробка бази даних програмного забезпечення для розпізнавання людських облич	43
3.4 Проєктування інтерфейсу програмного забезпечення для розпізнавання людських облич	45
3.5 Висновки за розділом 3	47
4 Експлуатація, тестування та експериментальне дослідження програми	49
4.1 Призначення й умови застосування програми	49
4.2 Характеристики програми для розпізнавання людських облич	50
4.3 Інструкція по експлуатації програми	51
4.3.1 Звернення до програми	51

4.3.2 Вхідні й вихідні дані	51
4.3.3 Повідомлення	52
4.4 Виконання програмного забезпечення для розпізнавання людських облич	52
4.5 Тестування програмного забезпечення для розпізнавання людських облич	54
4.6 Висновки за розділом 4	54
Висновки	56
Перелік джерел посилань	59
Додаток А Технічне завдання	61
Додаток Б Фрагмент тексту програми	65
Додаток В Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

БД – база даних;

ПЗ – програмне забезпечення;

РЛО – розпізнавання людських облич.

Вступ

Актуальність програмного забезпечення для розпізнавання людських облич є досить високою. Таке програмне забезпечення використовується у галузях безпеки, підтримки персоналізації, медицини, освіти та дистанційної роботи, фінансових технологіях та ін. [1]-[7].

Програмне забезпечення для розпізнавання людських облич використовується для автоматичного виявлення осіб у місцях з високим рівнем безпеки, наприклад, в аеропортах, банках чи на державних об'єктах. Розпізнавання облич за допомогою програмних засобів також використовується поліцією та службами безпеки для пошуку підозрюваних у великих базах даних [1], [2].

Програмні засоби з підтримкою розпізнавання облич дозволяють надавати персоналізовані пропозиції клієнтам на основі вікових або емоційних ознак. У медичній галузі програмне забезпечення для розпізнавання людських облич може використовуватися для контролю стану втоми або сонливості у водіїв і працівників у небезпечних умовах, виявлення генетичних та інших захворювань за характерними рисами обличчя, моніторингу емоційного стану пацієнтів у терапевтичних цілях [1], [3].

В освітній галузі та дистанційній роботі програмне забезпечення для розпізнавання людських облич дозволяє виконувати контроль присутності та участі студентів під час онлайн навчання, перевірку особистості на іспитах для уникнення шахрайства, а також контролювати співробітників, що працюють дистанційно [2], [4].

Проте програмні засоби для розпізнавання людських облич характеризуються низкою недоліків, які обмежують їх ефективність і викликають етичні та технічні дискусії. Деяке програмне забезпечення для розпізнавання людських облич можуть використовувати алгоритми, які забезпечують низьку точність розпізнавання у складних умовах, зокрема, при

поганому освітленні, частковому закритті обличчя, різних ракурсах або емоційних виразах обличчя. Деякі алгоритми, що використовуються у програмному забезпеченні для розпізнавання облич, можуть працювати менш ефективно для людей із певними етнічними або демографічними характеристиками, що може спричиняти дискримінацію. Крім того, деякі програмні системи розпізнавання облич можуть потребувати потужного апаратного забезпечення для обробки зображень і даних у реальному часі, вимагати регулярного оновлення, адаптації до нових умов і тестування, залежати від якості та кількості даних для навчання [1]-[7].

Тому актуальною є розробка програмного забезпечення для розпізнавання людських облич. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для розпізнавання людських облич.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання образів, зокрема, людських облич;
- здійснити проєктування програмного забезпечення для розпізнавання людських облич;
- створити програмне забезпечення для розпізнавання людських облич;
- виконати тестування розробленого програмного забезпечення для розпізнавання людських облич.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для розпізнавання образів

Розпізнавання образів являє собою процес ідентифікації структур за допомогою алгоритмів машинного навчання. Його суть полягає у класифікації даних, що базується на попередньо отриманих знаннях або статистичних характеристиках, які були виділені з наявних зразків чи їхнього представлення. Одним із ключових моментів такого підходу є широкий спектр можливих застосувань [1].

Стандартний процес розпізнавання передбачає попередню обробку даних для їх перетворення у форму, зручну для аналізу алгоритмами. Результатом цього є класифікація або групування виявлених шаблонів [1].

У випадку класифікації кожному шаблону присвоюється відповідна мітка класу. Це відбувається на основі узагальненої моделі, створеної з навчальних даних або за допомогою експертних знань. Класифікація є основним підходом у контрольованому навчанні [1].

Кластеризація, навпаки, спрямована на поділ даних на групи, що сприяє прийняттю рішень у певних задачах. Цей підхід використовується у неконтрольованому навчанні для визначення подібностей між об'єктами [1].

Характеристики, які описують об'єкти, можуть бути представлені безперервними, дискретними або двійковими змінними. Вони дозволяють кількісно оцінити важливі властивості об'єкта на основі одного чи кількох вимірювань. Наприклад, при розгляді обличчя окремими характеристиками будуть його очі, вуха чи ніс. Сукупність таких ознак формується у вектор характеристик, який використовується для подальшого аналізу [1].

Розпізнавання образів належить до галузі машинного навчання, де основною метою є розробка методів для автоматичного виявлення та ідентифікації закономірностей у даних. Процес базується на створенні моделей

та алгоритмів, здатних аналізувати інформацію і робити прогнози або приймати рішення на основі отриманих результатів [1].

Важливими аспектами цього підходу є представлення даних у вигляді характеристик, які обираються відповідно до специфіки завдання. Саме кодування даних відіграє ключову роль у загальному успіху системи. Для оцінки схожості між різними наборами даних використовується ймовірнісний підхід, який забезпечує гнучкість у порівнянні структур інформації [1].

Моделі, що застосовуються в таких системах, варіюються від лінійних до нелінійних і ймовірнісних, залежно від типу даних та поставленої задачі. Їхній вибір визначається відповідністю специфіці проблеми, яку вирішує система. Крім того, для перевірки ефективності моделей використовуються оцінювальні параметри, що дозволяють порівнювати різні алгоритми і визначати найкращий варіант [1].

Перед початком аналізу проводиться попередня обробка, яка включає очищення, фільтрацію або перетворення даних для підготовки їх до подальшого аналізу. У процесі також здійснюється вибір найбільш релевантних характеристик, що сприяє підвищенню точності і зниженню обчислювальної складності [1].

Для підвищення продуктивності важливим є оптимальне налаштування системи на основі навчальних даних. Якість навчального набору безпосередньо впливає на загальну ефективність алгоритму чи моделі, що використовуються. У підсумку процес розпізнавання спрямований на побудову інструментів, здатних успішно обробляти великі обсяги даних, виявляти закономірності та забезпечувати прийняття рішень у різних прикладних задачах [1].

Розпізнавання людських облич (РЛО) є процесом автоматичної ідентифікації або верифікації особи на основі аналізу її обличчя за допомогою алгоритмів комп'ютерного зору та машинного навчання. Ця технологія широко

застосовується у сфері безпеки, мобільних пристроях, системах контролю доступу та персоналізації [1], [2].

Основна ідея розпізнавання людських облич полягає у виділенні унікальних характеристик обличчя, таких як форма, розташування очей, носа, рота, текстура шкіри або вирази, і перетворення цих даних у цифровий формат. Алгоритми РЛО аналізують отриману інформацію, порівнюючи її з базою збережених зразків для визначення особи [1], [2].

Процес розпізнавання зазвичай складається з кількох етапів: виявлення обличчя: знаходження облич на фото чи у відео; попередня обробка: налаштування освітлення, вирівнювання та масштабування зображення; виділення ознак: аналіз ключових точок обличчя, таких як положення очей, носа, форми губ; порівняння: зіставлення даних із наявними шаблонами для ідентифікації або верифікації [1], [2].

Сучасні системи розпізнавання облич використовують методи глибокого навчання, наприклад, нейронні мережі, які забезпечують високу точність навіть за умов низької якості зображення або змін у зовнішності людини. Популярні бібліотеки та платформи, такі як Face++, Kairos або HyperVerge, пропонують засоби для інтеграції технології у різні програмні застосунки [1], [2].

Технологія розпізнавання облич отримала широке поширення, але викликає занепокоєння щодо конфіденційності, безпеки даних та етичних аспектів використання [1], [2].

Зростання значення людського фактора в сучасних технологіях формує потребу у впровадженні біометричних систем для ідентифікації та перевірки особи. Серед таких систем використовуються як фізіологічні статичні ознаки, наприклад, відбитки пальців, структура райдужної оболонки або малюнок долоні, так і поведінкові характеристики, що включають ходу, підпис, мовленнєві особливості, динаміку рухів обличчя та інші аспекти, відомі під назвою м'якої біометрії [1], [2].

Обличчя відіграє ключову роль у біометричній ідентифікації завдяки широкому спектру застосувань: від забезпечення безпеки й підтримки правопорядку до використання в медичній галузі, освіті, фінансових операціях, маркетингу, індустрії розваг і технологіях взаємодії людини з комп'ютером. Воно несе інформацію про індивідуальність, вік, стать, етнічну належність, а також емоційний та психічний стан, що знаходить відображення у виразах обличчя. Дослідження в цій галузі поєднують знання з психології та технічних наук [1], [2].

На відміну від багатьох інших біометричних характеристик, розпізнавання облич не потребує активної участі людини, що дозволяє застосовувати цю технологію непомітно, наприклад, у системах відео спостереження. Окрім того, аналіз обличчя може базуватися як на статичних рисах, так і на їх динамічних змінах, що розширює можливості його використання для поведінкової біометрії [1], [2].

Розпізнавання облич в умовах, де відсутні суворі обмеження, є складним завданням через такі фактори, як варіації положення голови, освітлення, вікові зміни чи зміни у виразах. Додаткові труднощі створюються змінами зовнішності, наприклад, через макіяж, зачіску або використання аксесуарів, таких як окуляри чи шарфи. Виявляється, що схожість між людьми, зокрема у випадках родичів або близнюків, також ускладнює процес розпізнавання [1], [2].

Хоча людський мозок здатний легко й ефективно розпізнавати обличчя, для комп'ютерних систем це завдання залишається технічно складним. Візуальна обробка облич у людей ґрунтується на складних нейронних механізмах, що аналізують як статичні, так і динамічні характеристики з урахуванням контексту. Дослідження у сфері психології та штучних нейронних мереж вивчають особливості сприйняття облич, зокрема використання цілісних уявлень або аналіз окремих ознак, деякі з яких виявляються домінуючими. Сприйняття облич відбувається завдяки низці незалежних

процесів, які дозволяють визначати вік, стать, емоції чи інші характеристики через фізичні риси. Ця модель обробки забезпечує здатність розпізнавати обличчя навіть за змінних умов. Крім того, існують дослідження, що зосереджуються на важливості окремих рис, таких як очі чи рот, для ефективного розпізнавання, а також на впливі освітлення чи руху на процес сприйняття знайомих облич [2].

Розвиток технологій розпізнавання обличч розпочався у середині двадцятого століття, а автоматизація цього процесу стала об'єктом досліджень наприкінці відповідного століття. Початкові методи базувалися на аналізі відстаней між ключовими точками обличчя. Прогрес апаратного забезпечення та зростання інтересу до технологій безпеки даних стимулювали активний розвиток цієї галузі. Еволюція підходів до розпізнавання обличч відбувалася в кілька етапів. Перший етап, що базувався на цілісному аналізі, передбачав використання методів для зменшення розмірності ознак обличчя. Згодом набули популярності локальні методи, які використовували створені вручну дескриптори, такі як локальні бінарні шаблони. Останні роки розвиток перейшов до навчання локальних дескрипторів за допомогою дискримінантних фільтрів. Зрештою, технології глибокого навчання, які здобули популярність після успіху мережі AlexNet, докорінно змінили підхід до розпізнавання, досягнувши рівня продуктивності, близького до людського, на великих наборах даних [2].

Процес розпізнавання обличчя передбачає визначення чи підтвердження особи за характеристиками обличчя. Для цього використовуються методи аналізу та порівняння унікальних рис, притаманних зовнішності кожної людини [3].

На початковому етапі здійснюється виявлення обличчя на фото чи відео, що дозволяє визначити його місцезнаходження у зображенні. Наступний крок полягає у цифровій обробці, під час якої риси обличчя перетворюються

на математичні дані. Завершальний етап процесу РЛО зводиться до порівняння отриманих даних із базами, щоб встановити особу або підтвердити її [3].

Серед усіх біометричних підходів розпізнавання облич вважається найбільш природним, оскільки саме обличчя найчастіше використовується людьми для взаємного впізнавання. Алгоритми аналізують такі риси, як відстань між очима, форма підборіддя, контури носа чи вилиць, створюючи унікальний цифровий підпис. Цей підпис зіставляється з уже наявними даними для пошуку збігів [3].

Цифрові технології дозволяють виконувати ідентифікацію чи автентифікацію особи навіть у складних умовах, наприклад, у натовпі або динамічному середовищі. Подібні методи вже інтегровані в багато пристроїв, таких як сучасні смартфони, і знаходять застосування в різних галузях – від безпеки до персоналізованих сервісів [3].

Інші види біометрії, включаючи аналіз відбитків пальців, райдужної оболонки чи голосу, також активно використовуються. Однак технології, що працюють з розпізнаванням облич, пропонують високу точність і швидкість роботи, роблячи їх незамінними у сучасних умовах [3].

Програмне забезпечення розпізнавання обличчя являє собою інструмент, що дозволяє ідентифікувати або підтверджувати особу людини за допомогою аналізу її обличчя. Таке програмне забезпечення використовує алгоритми, які досліджують особливості обличчя, такі як відстань між очима, форма підборіддя чи контури вилиць, створюючи унікальний цифровий підпис. Потім отримані дані зіставляються з базами, де зберігаються відомі зображення, для пошуку збігів [3], [4].

Сучасні програмні рішення на основі штучного інтелекту забезпечують значні переваги в цій галузі. Технологія сприяє підвищенню рівня безпеки, мінімізує ймовірність крадіжки персональних даних і забезпечує зручність ідентифікації в різних сферах застосування [3], [4].

Процес розпізнавання обличчя проходить кілька етапів. Спочатку здійснюється захоплення зображення, що може бути виконано як у реальному часі за допомогою камер, так і на основі вже наявних фото. Далі відбувається виділення обличчя з усього зображення, а потім аналіз його характерних рис. Ці дані переводяться в цифрову форму й порівнюються з уже збереженими записами для визначення збігів [3], [4].

Ця технологія активно застосовується в багатьох галузях. У фінансовій сфері вона використовується для захисту від шахрайства та спрощення процесів ідентифікації. У транспортній безпеці забезпечується ефективна перевірка пасажирів у аеропортах. В охороні здоров'я технологія допомагає у правильній ідентифікації пацієнтів для забезпечення належного лікування. У телекомунікаціях вона оптимізує процес верифікації клієнтів, а в економіці послуг забезпечує перевірку надійності постачальників [3], [4].

Серед важливих характеристик програмного забезпечення для розпізнавання обличчя виділяються точність алгоритмів, здатність обробляти дані в реальному часі, інтеграція з іншими системами, масштабованість і зручність використання. Крім того, для забезпечення надійності та етичності програмне забезпечення має зменшувати упередженість, підтримувати персоналізовані налаштування і відповідати стандартам конфіденційності та захисту даних [3], [4].

Розпізнавання обличчя, як одна з біометричних технологій, представляє собою сучасний підхід до ідентифікації особистості та забезпечення безпеки. Основний принцип роботи цієї системи полягає у використанні унікальних рис обличчя для встановлення або підтвердження особи [3], [4].

Технологія РЛО входить до складу біометричних методів, які включають також сканування відбитків пальців, аналіз райдужної оболонки ока та ідентифікацію за голосом. Усі ці технології об'єднує спільна мета – забезпечити точну ідентифікацію людини на основі її фізичних характеристик. Розпізнавання обличчя виділяється тим, що спирається на специфічну

конфігурацію рис обличчя, що робить його ефективним і зручним інструментом [3], [4].

Однією з ключових переваг програмного забезпечення для РЛО є можливість виконання ідентифікації на відстані без необхідності прямої взаємодії з людиною. Така властивість значно спрощує використання технології та дозволяє застосовувати її у різних сферах – від розблокування телефонів до посилення заходів безпеки в аеропортах [3], [4].

Щодо переваг, розпізнавання обличчя демонструє значний потенціал у підвищенні рівня безпеки. Завдяки високій точності, система дозволяє забезпечити контроль доступу та знижує ризики шахрайства чи несанкціонованих дій. У місцях із посиленими заходами безпеки, таких як фінансові установи чи державні будівлі, ця технологія додає важливий рівень захисту. Окрім цього, вона легко інтегрується з уже існуючими системами, такими як перевірка документів чи інші біометричні рішення, що розширює можливості її впровадження у різних галузях [3], [4].

Точність у процесі ідентифікації також є важливим аспектом. Завдяки використанню передових алгоритмів і методів машинного навчання, розпізнавання обличчя забезпечує високу достовірність результатів, що має особливе значення в ситуаціях, де помилки можуть призвести до серйозних наслідків, наприклад, у фінансовій чи правоохоронній сферах [3], [4].

Технологія розпізнавання обличчя включає кілька підходів, які забезпечують її ефективність і гнучкість. Серед них використання згорткових нейронних мереж, які застосовуються для аналізу зображень за допомогою спеціальних алгоритмів, і методів зменшення розмірності даних, які дозволяють виділяти ключові ознаки обличчя. Також існують тривимірні моделі, що враховують структуру черепа, та аналіз текстури шкіри, який базується на вивченні деталей, таких як родимки чи колір шкіри. Додатково застосовуються теплові камери для виявлення температурних особливостей

обличчя, а також передові рішення, такі як FaceNet, які забезпечують високу точність завдяки спеціалізованим вбудованим алгоритмам [3], [4].

Технологія розпізнавання обличчя включає кілька взаємопов'язаних етапів, заснованих на використанні сучасних алгоритмів та підходів. Спочатку система ідентифікує область обличчя, зчитуючи біометричні характеристики. Найкращі результати досягаються за прямого погляду в камеру, хоча сучасні методи дозволяють працювати і з деякими відхиленнями. Після визначення обличчя створюється зображення, яке аналізується з урахуванням ключових точок [4], [5].

Далі ця інформація перетворюється в цифрові дані, що утворюють унікальний код — відбиток обличчя. Завдяки цьому коду відбувається порівняння з існуючою базою даних для підтвердження особи. У разі відповідності система надає інформацію, пов'язану з ідентифікованим обличчям, наприклад, ім'я чи інші персональні дані [5], [6].

Сфера застосування такої технології є досить широкою. У безпеці вона допомагає виявляти підозрюваних та забезпечувати охорону об'єктів, а в медицині — покращувати діагностику і лікування пацієнтів. У роздрібній торгівлі використовується для персоналізації сервісу та запобігання крадіжкам, тоді як у фінансовій сфері допомагає гарантувати безпеку транзакцій. У транспорті технологія спрощує процедури реєстрації та посадки, а в телефонах забезпечує швидкий доступ і захист даних [6], [7].

При оцінюванні таких систем враховуються кілька ключових аспектів. Одним із них є перевірка живучості, яка захищає від спроб обману за допомогою фото або відео. Важливу роль відіграють обрані алгоритми, такі як нейронні мережі чи власні обличчя, а також можливість системи адаптуватися до нових даних. Щоб уникнути упереджень, моделі повинні тренуватися на різноманітних наборах даних. Крім того, сучасні виклики, як-от дипфейки, вимагають особливої уваги до достовірності інформації. Точність

розпізнавання залишається пріоритетом навіть за умов поганого освітлення чи інших складнощів [6], [7].

Розпізнавання емоцій на основі аналізу обличчя передбачає визначення виразів, які вказують на такі емоції, як радість, сум, здивування, страх, гнів або складні комбінації цих почуттів. Цей процес реалізується через програмне забезпечення для обробки зображень, яке може працювати як із зображеннями в реальному часі, так і зі статичними кадрами [6], [7].

Алгоритми розпізнавання емоцій включають кілька послідовних етапів. Спочатку визначається обличчя в полі зору системи, після чого аналізуються його мімічні особливості. На основі цих даних визначається емоційний стан, який відображають вирази обличчя [6], [7].

На відміну від технологій розпізнавання облич, які спрямовані на встановлення особистості, розпізнавання емоцій фокусується виключно на інтерпретації внутрішнього стану людини. Вирази обличчя аналізуються за допомогою геометричних параметрів, текстурних характеристик або тривимірних моделей, створених на основі оброблених зображень [6], [7].

Процес розпізнавання обличчя складається з трьох основних етапів: пошуку обличчя, його аналізу та подальшого встановлення ідентичності. Перший етап передбачає визначення місцезнаходження обличчя на зображенні. За допомогою технології комп'ютерного зору обличчя можна виявляти як у фронтальному вигляді, так і у профільних ракурсах, навіть якщо на зображенні присутні кілька людей. Комп'ютерний зір дозволяє машині ефективно ідентифікувати об'єкти, використовуючи штучний інтелект для аналізу, класифікації та інтерпретації даних із зображень різного формату, включно з одиничними фото, відеозаписами, багатокамерними знімками чи тривимірними моделями [6], [7].

На наступному етапі виконується аналіз структури обличчя. Система вивчає геометричні особливості, такі як відстань між очима, довжина від чола до підборіддя, форма або контур губ, і на основі цих параметрів створює

цифрову модель – унікальний «відбиток обличчя». Цей відбиток дає змогу відрізнати одне обличчя від іншого та навіть дозволяє виконувати цифрову реконструкцію вигляду людини [6], [7].

На завершальному етапі здійснюється ідентифікація, яка полягає у порівнянні облич між собою. Наприклад, система може перевірити відповідність зображення, отриманого через мобільний пристрій, до фотографії у посвідченні або іншому документі. У деяких випадках технологія також може підтверджувати, що зображення не збігається з раніше зареєстрованими обличчями [6], [7].

Хоча в лабораторних умовах алгоритми розпізнавання демонструють майже ідеальну точність, їхня ефективність може знижуватися в реальних ситуаціях через непередбачувані фактори. Висока точність можливіша при оптимальних умовах, таких як рівномірне освітлення, чіткість зображення та контрольоване оточення. Проте старіння людини також може ускладнювати зіставлення з давніми фотографіями, що впливає на точність аналізу [6], [7].

Визначено, що розпізнавання людських облич є процесом автоматичної ідентифікації або верифікації особи на основі аналізу її обличчя за допомогою алгоритмів комп'ютерного зору та машинного навчання, ця технологія широко застосовується у сфері безпеки, мобільних пристроях, системах контролю доступу та персоналізації. Відзначено, що програмне забезпечення розпізнавання обличчя являє собою інструмент, що дозволяє ідентифікувати або підтверджувати особу людини за допомогою аналізу її обличчя. Таке програмне забезпечення використовує алгоритми, які досліджують особливості обличчя, такі як відстань між очима, форма підборіддя чи контури вилиць, створюючи унікальний цифровий підпис, потім отримані дані зіставляються з базами, де зберігаються відомі зображення, для пошуку збігів.

1.2 Аналіз програмного забезпечення для підтримки процесу розпізнавання людських облич

Проаналізуємо програмне забезпечення для підтримки процесу розпізнавання людських облич [8]-[14].

Програмне забезпечення Kairos (рис. 1.1) являє собою хмарну платформу для розпізнавання облич, яка пропонує інструменти для аналізу та ідентифікації осіб. Основний акцент цього програмного забезпечення зроблено на інтеграції з різними програмами та сервісами, а також на підтримці аналізу зображень і відео в реальному часі. Програмне забезпечення Kairos розроблене для використання в комерційних і корпоративних цілях, що дозволяє підприємствам автоматизувати процеси розпізнавання та відстеження [8], [10], [12].

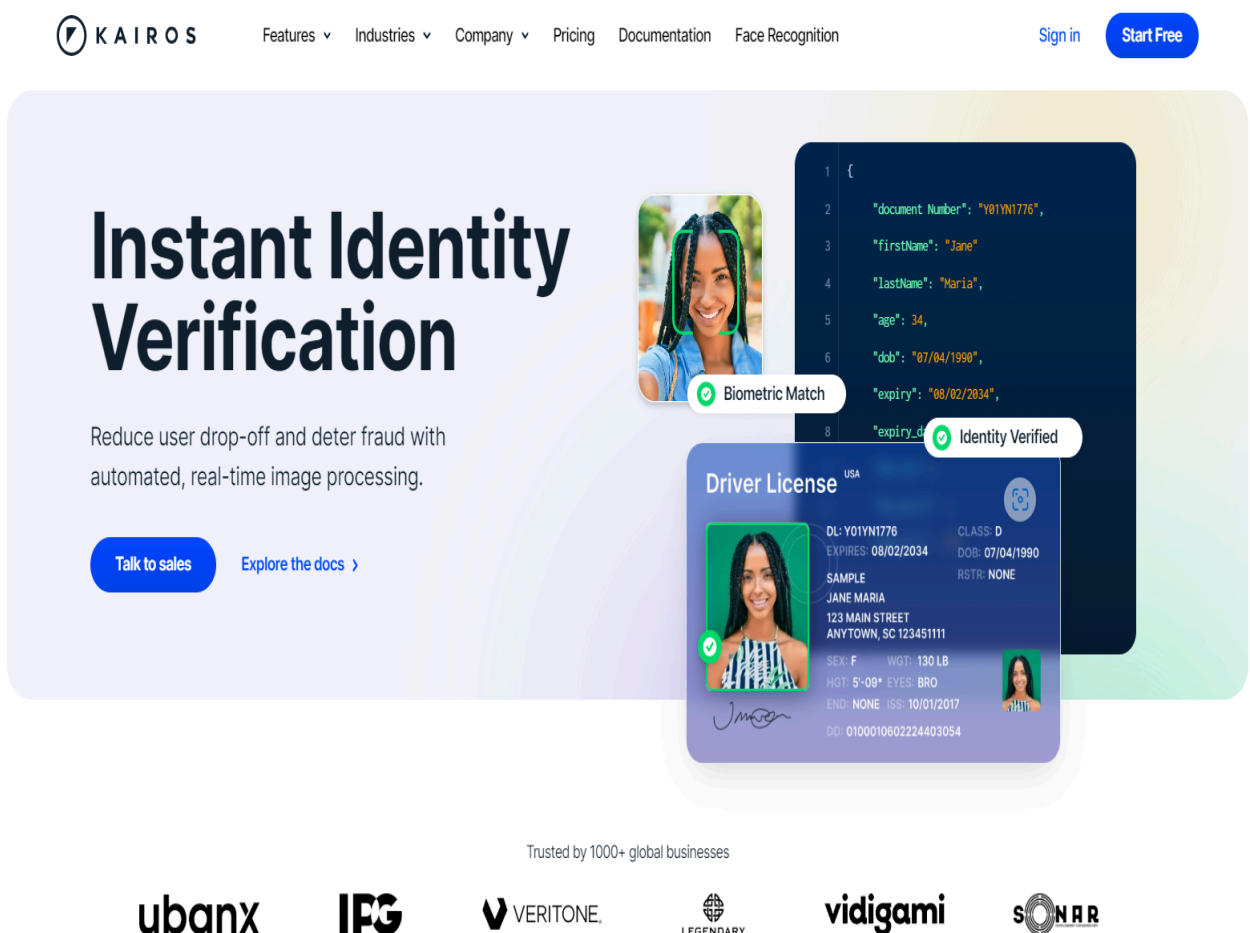


Рисунок 1.1 – Програмне забезпечення Kairos [12]

Однією з ключових функцій програмного забезпечення Kairos є ідентифікація облич, яка дозволяє системі зіставляти обличчя із записами в базі даних, забезпечуючи точну ідентифікацію. Також підтримується виявлення облич на зображеннях, що дозволяє визначати кількість осіб та їхні позиції. Kairos пропонує функцію аналізу демографічних характеристик, таких як вік, стать та етнічна приналежність, що робить програму корисною для маркетингових досліджень [8], [10], [12].

Програмне забезпечення Kairos забезпечує високу конфіденційність даних, оскільки пропонує можливість локального зберігання даних замість хмарних сервісів, якщо це необхідно. Kairos легко інтегрується з програмами, що розробляються на різних мовах програмування, таких як Python, JavaScript або Ruby [8], [10], [12].

Програмне забезпечення Kairos має широке застосування в різних сферах, включаючи безпеку, роздрібну торгівлю, банківські послуги та охорону здоров'я. Наприклад, у системах безпеки платформа може використовуватися для контролю доступу до приміщень або пристроїв, а в торгівлі — для аналізу поведінки клієнтів [8], [10], [12].

Гнучкість і адаптивність програмного забезпечення дозволяють налаштовувати функціонал відповідно до потреб користувача, що робить Kairos універсальним рішенням для роботи з технологіями розпізнавання облич [8], [10], [12].

Основними характеристиками програмного забезпечення Kairos є такі [8], [10], [12]:

- ідентифікація облич: програмне забезпечення Kairos забезпечує можливість зіставлення облич із базою даних для точної ідентифікації осіб, що важливо для систем безпеки чи контролю доступу [8], [10], [12];

– виявлення облич: програмне забезпечення Kairos дозволяє знаходити обличчя на зображеннях чи у відеопотоці, визначаючи їхні положення, що корисно для аналізу зображень і відео [8], [10], [12];

– аналіз демографії: програма здатна визначати характеристики, такі як вік, стать та етнічну приналежність, що використовується для маркетингових досліджень або персоналізації [8], [10], [12];

– робота в реальному часі: програмне забезпечення Kairos підтримує обробку даних з відео потоків у режимі реального часу, що робить програму ефективною для моніторингу та миттєвого реагування [8], [10], [12];

– конфіденційність даних: програмне забезпечення Kairos забезпечує можливість локального зберігання даних, що підвищує рівень безпеки та захисту конфіденційної інформації [8], [10], [12];

– інтеграція: програмне забезпечення Kairos легко інтегрується з програмами, написаними на різних мовах програмування, таких як Python, JavaScript, Ruby, що дозволяє адаптувати розроблюване програмне забезпечення до різних потреб [8], [10], [12];

– висока точність розпізнавання: використання алгоритмів глибокого навчання гарантує точність результатів навіть за складних умов, таких як низька якість зображень [8], [10], [12];

– масштабованість: програмне забезпечення Kairos підтримує роботу з великими обсягами даних і може бути адаптоване для потреб як малого бізнесу, так і великих організацій [8], [10], [12];

– кросплатформність: програмне забезпечення Kairos може працювати на різних операційних системах, забезпечуючи гнучкість і зручність для користувачів [8], [10], [12];

– адаптивність до різних галузей: програмне забезпечення Kairos може використовуватися в різних сферах, таких як безпека (контроль доступу), роздрібна торгівля (аналіз клієнтів), фінанси (верифікація особи), охорона здоров'я (ідентифікація пацієнтів) та інші [8], [10], [12].

Серед недоліків програмного забезпечення Kairos можна відзначити залежність від якості зображень, програма може демонструвати знижену точність при роботі з розмитими, темними або низькоякісними зображеннями, що обмежує її ефективність у складних умовах. Для забезпечення роботи з відео потоками або обробки великих обсягів даних за допомогою програмного забезпечення Kairos необхідні значні обчислювальні ресурси, що може бути проблемою для менш потужних систем. Крім того, можливе зниження точності при роботі з певними демографічними групами через обмеження у наборі даних для тренування алгоритмів [8], [10], [12].

Програмне забезпечення HyperVerge (рис. 1.2) представляє собою програмне забезпечення для розпізнавання облич, яке використовує сучасні алгоритми комп'ютерного зору та машинного навчання для ідентифікації та перевірки особистості. Програмне забезпечення HyperVerge створене для автоматизації завдань розпізнавання осіб у реальному часі та оптимізації бізнес-процесів у різних галузях. Програмне забезпечення HyperVerge активно використовується в банківській справі, фінансових технологіях, безпеці, страхуванні, а також у логістиці [8], [13].

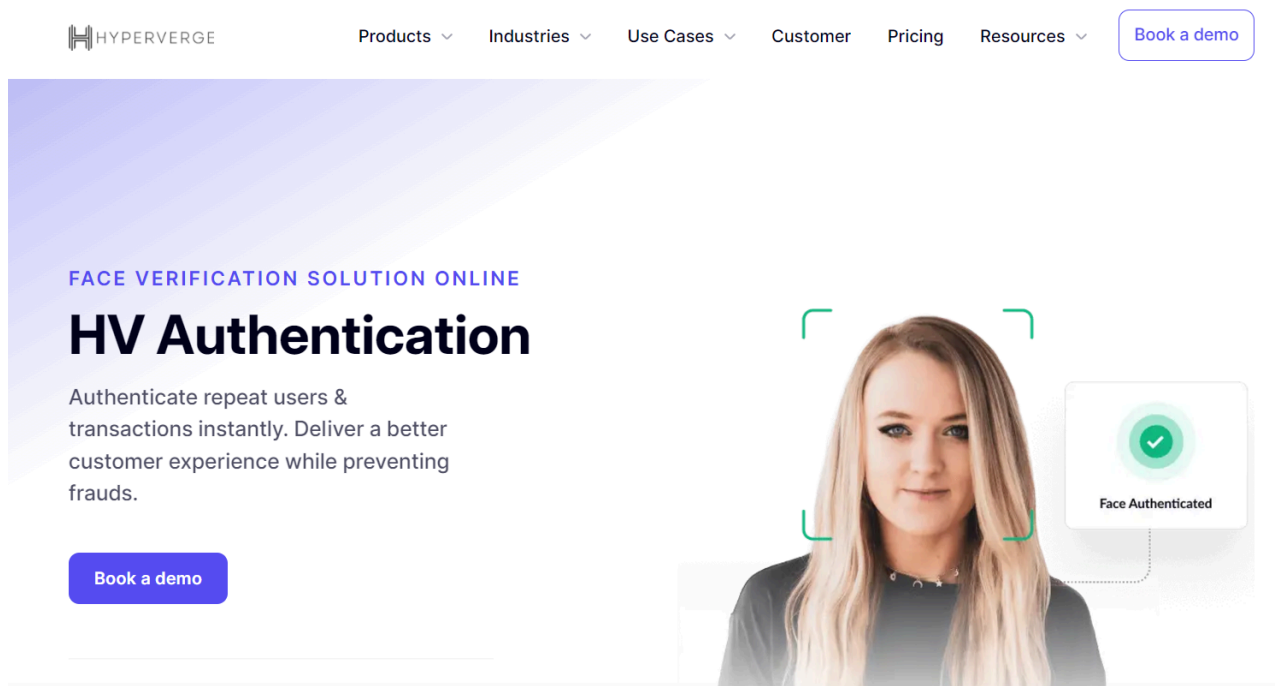


Рисунок 1.2 – Програмне забезпечення HyperVerge [13]

Основні можливості HyperVerge включають розпізнавання облич у фотографіях, відео потоках та зображеннях, отриманих у реальному часі. Система може ідентифікувати людей за їхніми біометричними характеристиками, а також перевіряти особисті документи на основі фотографій. Висока точність розпізнавання забезпечується за рахунок використання передових алгоритмів глибокого навчання [8], [13].

HyperVerge підтримує інтеграцію з іншими програмними системами, що робить його універсальним рішенням для бізнесу. Система оптимізована для роботи на хмарних платформах, але може функціонувати і в локальних середовищах, забезпечуючи безпеку даних [8], [13].

Однією з ключових переваг програмного забезпечення є його швидкість і точність. Програмне забезпечення HyperVerge здатно виконувати розпізнавання та перевірку у реальному часі навіть за умов низької якості зображень. Завдяки цьому рішення ефективно працює у різних сценаріях, включаючи ситуації з поганим освітленням або складним фоном [8], [13].

Програмне забезпечення також забезпечує підтримку багатofакторної ідентифікації, що підвищує рівень безпеки для користувачів. HyperVerge активно використовується для автоматизації процесів у банківському секторі, що дозволяє значно скоротити час перевірки документів та клієнтів [8], [13].

Таким чином, HyperVerge є потужним інструментом для впровадження сучасних технологій розпізнавання облич у бізнес-процеси, забезпечуючи високу продуктивність, точність і гнучкість [8], [13].

Програмне забезпечення HyperVerge має такі особливості [8], [13]:

- розпізнавання облич у реальному часі: HyperVerge забезпечує миттєве розпізнавання облич навіть у відеопотоках. Це дозволяє ефективно ідентифікувати осіб у динамічних умовах, наприклад, на камерах спостереження [8], [13];

- висока точність алгоритмів: програмне забезпечення HyperVerge використовує методи глибокого навчання, які забезпечують точне розпізнавання навіть за умов низької якості зображень [8], [13];

- біометрична ідентифікація: HyperVerge дозволяє визначати особу на основі біометричних даних, таких як особливості обличчя, що мінімізує ризики підробки чи помилкової ідентифікації [8], [13];

- інтеграція через інтерфейс прикладного програмування (Application Programming Interface, API): програмне забезпечення HyperVerge надає легкий доступ до функціоналу програми через API, що дозволяє інтегрувати технології розпізнавання облич в існуючі системи чи програми [8], [13];

- автоматизація процесів: програмне забезпечення HyperVerge активно використовується для автоматизації перевірки клієнтів у фінансових установах. Це скорочує час, необхідний для ідентифікації та верифікації клієнтів, і знижує витрати [8], [13];

- аналіз документів: HyperVerge може порівнювати фото користувача з його офіційними документами, такими як паспорти чи посвідчення особи, для підтвердження справжності даних [8], [13];

- швидкість обробки даних: програмне забезпечення HyperVerge оптимізовано для роботи з великими обсягами даних, забезпечуючи швидке виконання задач навіть за великого навантаження [8], [13];

- хмарна та локальна підтримка: HyperVerge може працювати як у хмарному середовищі, так і локально, що дозволяє компаніям обирати оптимальне середовище для своїх потреб, забезпечуючи при цьому високу безпеку даних [8], [13];

- багатофакторна ідентифікація: програмне забезпечення HyperVerge підтримує багатофакторну перевірку користувачів, що додає рівень безпеки та знижує ризик несанкціонованого доступу [8], [13];

– оптимізація під різні умови: програмне забезпечення HyperVerge добре працює в умовах поганого освітлення, незадовільної якості камер або складних фонових умов, що робить його універсальним інструментом [8], [13];

– підтримка різних галузей: програмне забезпечення широко використовується у банківській справі, фінансових технологіях, логістиці, безпеці та страхуванні, забезпечуючи гнучкість у впровадженні [8], [13].

Серед недоліків програмного забезпечення HyperVerge можна відзначити обмеження для нестандартних сценаріїв розпізнавання людських облич, алгоритми можуть демонструвати знижену точність у випадках з дуже складними або нестандартними умовами, наприклад, сильне затемнення, незвичайні ракурси або часткове закриття облич. При локальному розгортанні програмне забезпечення HyperVerge потребує значних обчислювальних ресурсів, що може бути викликом для компаній з обмеженою технічною інфраструктурою. Хоча HyperVerge забезпечує API, інтеграція з певними менш популярними мовами програмування може бути обмеженою, що ускладнює впровадження для деяких розробників [8], [13].

Програмне забезпечення Face++ (рис. 1.3) надає інструменти для обробки зображень і відео з використанням алгоритмів комп'ютерного зору та штучного інтелекту. Програмне забезпечення Face++ широко використовується в різних галузях, включаючи безпеку, фінанси, роздрібну торгівлю, розваги та охорону здоров'я [8], [14].

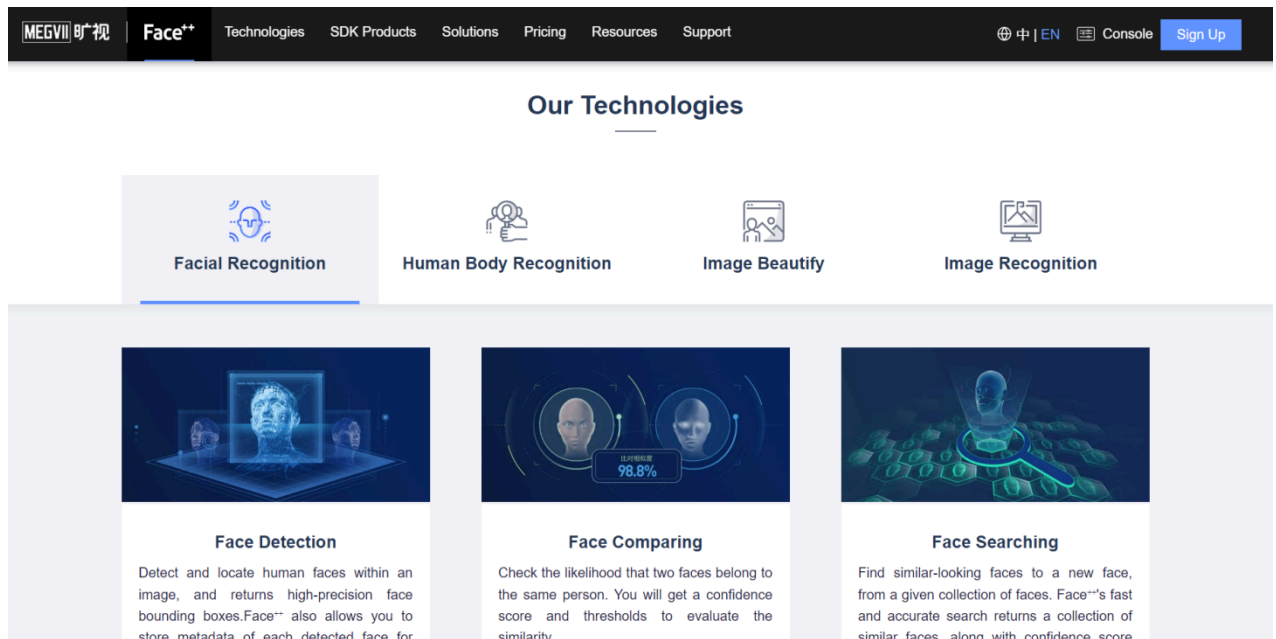


Рисунок 1.3 – Програмне забезпечення Face++ [14]

Програмне забезпечення Face++ надає широкий спектр можливостей у сфері розпізнавання облич. Програмне забезпечення дозволяє ідентифікувати обличчя на зображеннях і відео, аналізувати їхні характеристики та визначати унікальних осіб. Його алгоритми здатні визначати емоційний стан, стать, вік і навіть етнічну належність людини. Технологія програмного забезпечення Face++ також забезпечує точне визначення ключових точок обличчя, як-от положення очей, носа чи рота, що робить її корисною для створення застосунків із доповненою реальністю [8], [14].

Серед функцій Face++ є верифікація особи через порівняння облич із попередньо збереженими зразками. Крім того, технологія дозволяє здійснювати пошук людей у великих базах даних із високою швидкістю й точністю. Також вона здатна порівнювати два обличчя для визначення ступеня їх схожості. Окрім цього, Face++ підтримує розпізнавання об'єктів і сцен, таких як тварини, транспортні засоби чи інші елементи на зображеннях. Інтеграція через API забезпечує зручність у використанні та дозволяє розробникам легко вбудовувати її в інші програми [8], [14].

Face++ вирізняється високою точністю завдяки використанню глибоких нейронних мереж і регулярним вдосконаленням алгоритмів. Багатомовний інтерфейс платформи сприяє її доступності для користувачів із різних країн [8], [14].

Технологію активно застосовують у сфері безпеки для відео спостереження й ідентифікації осіб, зокрема злочинців. У фінансовому секторі вона використовується для підтвердження особи під час здійснення мобільних транзакцій. Крім того, Face++ знаходить застосування у розважальних додатках, наприклад, для створення фільтрів або аналізу емоцій. Завдяки своїй функціональності, точності та широкій сфері застосування, Face++ є одним із провідних рішень на ринку [8], [14].

Програмне забезпечення Face++ має такі особливості [8], [14]:

- розпізнавання облич: програмне забезпечення Face++ здатне ідентифікувати людські обличчя на зображеннях і відео з високою точністю. Алгоритми використовуються для виявлення розташування облич і обробки їхніх характеристик [8], [14];

- аналіз емоцій, віку та статі: програмне забезпечення Face++ визначає емоційний стан людини, приблизний вік і стать. Це корисно для аналітики споживачів, віртуальних помічників або персоналізації сервісів [8], [14];

- визначення ключових точок обличчя: програмне забезпечення Face++ виділяє ключові точки обличчя, включаючи положення очей, брів, носа та рота. Це робить платформу ідеальною для додатків з доповненою реальністю або обробкою відео [8], [14];

- ідентифікація особи: технологія дозволяє зіставляти обличчя із збереженими зразками, забезпечуючи швидку ідентифікацію користувачів для систем безпеки або фінансових додатків [8], [14];

- виявлення схожості облич: програмне забезпечення порівнює два зображення для оцінки схожості облич. Це корисно в системах контролю доступу або при виявленні дублюючих записів у базах даних [8], [14];

- пошук у базах даних: Face++ дозволяє виконувати пошук облич у великих базах даних із високою швидкістю, що застосовується у правоохоронній діяльності або в маркетингових дослідженнях [8], [14];

- розпізнавання об'єктів і сцен: окрім облич, система може аналізувати зображення для ідентифікації об'єктів, таких як транспортні засоби, тварини або сцени, що робить її корисною для відео аналітики [8], [14];

- інтеграція через API: Face++ пропонує зручні API для інтеграції в інші програмні продукти. Розробники можуть використовувати цей функціонал для створення кастомізованих рішень [8], [14];

- підтримка відео потоку: алгоритми дозволяють працювати з поточним відео в реальному часі, що використовується у відеоспостереженні або інтерактивних додатках [8], [14];

- кросплатформеність: Face++ підтримує різні операційні системи, включаючи Windows, macOS, Android та iOS, що дозволяє інтегрувати її у додатки для будь-яких платформ [8], [14];

- масштабованість: система здатна обробляти великі обсяги даних без втрати продуктивності, що є важливим для корпоративних клієнтів [8], [14];

- підтримка багатомовного інтерфейсу: інструмент надає інтерфейс та документацію кількома мовами, що робить його зручним для глобальних користувачів [8], [14];

- безпека даних: особлива увага приділяється захисту конфіденційної інформації, що дозволяє використовувати платформу в проєктах, де необхідна висока ступінь безпеки даних [8], [14].

Серед недоліків програмного забезпечення Face++ можна відзначити залежність від якості зображень, програма може демонструвати знижену точність при роботі з розмитими, темними або низькоякісними зображеннями, що обмежує її ефективність у складних умовах. Для забезпечення роботи з відео потоками або обробки великих обсягів даних за допомогою програмного забезпечення Kairos необхідні значні обчислювальні ресурси, що може бути

проблемою для менш потужних систем. Крім того, можливе зниження точності при роботі з певними демографічними групами через обмеження у наборі даних для тренування алгоритмів [8], [14].

Порівняльну характеристику програмного забезпечення для підтримки процесу розпізнавання людських облич наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для підтримки процесу розпізнавання людських облич

Критерій порівняння	Kairos [12]	HyperVerge [13]	Face++ [14]
Висока точність розпізнавання	+—	+	+
Інтеграція з іншими системами	+	+—	+—
Безпека даних	+	+—	—
Стійкість до поганих умов розпізнавання	+—	+—	+—
Ефективність обробки великих наборів даних	+—	+	+
Залежність від доступу до Інтернету	+—	+—	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для підтримки процесу розпізнавання людських облич. Проте програмні засоби для розпізнавання людських облич характеризуються низкою недоліків, які обмежують їх ефективність і викликають етичні та технічні дискусії. Деяке програмне забезпечення для розпізнавання людських облич можуть використовувати алгоритми, які забезпечують низьку точність розпізнавання у складних умовах, зокрема, при поганому освітленні, частковому закритті обличчя, різних ракурсах або емоційних виразах обличчя. Деякі алгоритми, що

використовуються у програмному забезпеченні для розпізнавання облич, можуть працювати менш ефективно для людей із певними етнічними або демографічними характеристиками, що може спричиняти дискримінацію. Крім того, деякі програмні системи розпізнавання облич можуть потребувати потужного апаратного забезпечення для обробки зображень і даних у реальному часі, вимагати регулярного оновлення, адаптації до нових умов і тестування, залежати від якості та кількості даних для навчання. Тому актуальною є розробка програмного забезпечення для розпізнавання людських облич.

При розробці програмного забезпечення для розпізнавання людських облич необхідно забезпечити такі функціональні вимоги:

- можливість введення файлу з графічним зображенням для розпізнавання обличчя на ньому;
- підтримка можливості розпізнавання особи за обличчям на зображенні;
- можливість виведення інформації про розпізнану особу;
- підтримка авторизації користувачів;
- підтримка можливості збереження додаткової інформації (ім'я, стать, адреси проживання та ін.) про осіб у базі даних з метою надання відповідної інформації як наслідок розпізнавання обличчя.

1.3 Висновки за розділом 1

Визначено, що розпізнавання людських облич є процесом автоматичної ідентифікації або верифікації особи на основі аналізу її обличчя за допомогою алгоритмів комп'ютерного зору та машинного навчання, ця технологія широко застосовується у сфері безпеки, мобільних пристроях, системах контролю доступу та персоналізації. Відзначено, що програмне забезпечення розпізнавання обличчя являє собою інструмент, що дозволяє ідентифікувати

або підтверджувати особу людини за допомогою аналізу її обличчя. Таке програмне забезпечення використовує алгоритми, які досліджують особливості обличчя, такі як відстань між очима, форма підборіддя чи контури вилиць, створюючи унікальний цифровий підпис, потім отримані дані зіставляються з базами, де зберігаються відомі зображення, для пошуку збігів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу розпізнавання людських облич. Проте програмні засоби для розпізнавання людських облич характеризуються низкою недоліків, які обмежують їх ефективність і викликають етичні та технічні дискусії. Деяке програмне забезпечення для розпізнавання людських облич можуть використовувати алгоритми, які забезпечують низьку точність розпізнавання у складних умовах, зокрема, при поганому освітленні, частковому закритті обличчя, різних ракурсах або емоційних виразах обличчя. Деякі алгоритми, що використовуються у програмному забезпеченні для розпізнавання облич, можуть працювати менш ефективно для людей із певними етнічними або демографічними характеристиками, що може спричиняти дискримінацію. Крім того, деякі програмні системи розпізнавання облич можуть потребувати потужного апаратного забезпечення для обробки зображень і даних у реальному часі, вимагати регулярного оновлення, адаптації до нових умов і тестування, залежати від якості та кількості даних для навчання. Тому актуальною є розробка програмного забезпечення для розпізнавання людських облич.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання людських облич.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для розпізнавання людських облич було використано мову програмування Python [15], [16].

Python є високорівневою об'єктно-орієнтованою мовою програмування з інтерпретованим підходом, що забезпечує потужні можливості для розробки сценаріїв. Завдяки високорівневим структурам даних, динамічній семантиці та можливостям інтеграції, мова програмування Python забезпечує швидкість розробки програм і розширення функціональності. Мова підтримує модульну архітектуру, що сприяє багаторазовому використанню коду та модулів. Інтерпретатор разом із загальними бібліотеками доступний для основних платформ і слугує основою для роботи. Python підтримує різноманітні стилі програмування, зокрема об'єктно-орієнтований, процедурний, функціональний та структурний [15], [16].

До основних переваг цієї мови належать чистий синтаксис, що полегшує роботу зі змінними, а також інфраструктура, яка включає підтримку компіляторів різних мов. Стандартний дистрибутив містить численні функціональні модулі, зокрема для розробки графічного інтерфейсу. Інтерактивний режим мови сприяє моделюванню та вирішенню простих задач. У стандартний дистрибутив входить функціонал IDLE, що написаний на Python і поєднує простоту з потужністю. Мова програмування Python також здатна вирішувати складні математичні задачі, підтримує роботу з комплексними числами, цілими значеннями та пропонує функціонал потужного калькулятора. Мова має відкритий код, що дозволяє користувачам вносити власні зміни [15], [16].

Ця мова забезпечує доступ до великої кількості бібліотек і платформ, орієнтованих на машинне навчання та комп'ютерний зір. Такі інструменти, як OpenCV, dlib, TensorFlow, PyTorch і Keras, створені для реалізації та навчання

моделей, що працюють із візуальними даними. Це суттєво зменшує затрати часу на розробку, оскільки основна увага приділяється побудові логіки програми, а не розробці складних алгоритмів [15], [16].

Python характеризується простотою синтаксису, що робить код зрозумілим і зручним для підтримки. Завдяки цьому нові члени команди можуть швидко включитися в роботу над проектом, що особливо важливо для великих команд і проектів з високою динамікою. Легкість у використанні також дозволяє ефективно створювати прототипи та тестувати нові ідеї, що є критичним для дослідницьких і комерційних ініціатив [15], [16].

Активна спільнота користувачів Python забезпечує доступ до різноманітних ресурсів, включаючи докладну документацію, приклади реалізацій, навчальні матеріали та консультаційну підтримку. Це значно полегшує процес вирішення складних задач під час розробки [15], [16]. Мова також забезпечує легку інтеграцію з іншими технологіями. Наприклад, обчислювальні модулі можна реалізувати на мовах C++ або Rust, а користувацький інтерфейс — у Python [15], [16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання людських облич наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання людських облич

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	Java
Бібліотеки для роботи з комп'ютерним зором	+	+-	+-
Простота синтаксису	+	+-	+
Активна спільнота	+	-	+-

Зручність для створення ПЗ для розпізнавання людських облич	+	—	—
Мультиплатформеність	+	+—	+

Мова програмування Python дозволяє створювати гнучкі та багатофункціональні системи. Крім того, Python підтримує роботу в різних операційних системах із мінімальними модифікаціями коду. Така універсальність сприяє ширшому використанню програмного забезпечення серед різноманітних груп користувачів [15], [16].

Отже, для реалізації програмного забезпечення для розпізнавання людських облич обрано мову програмування Python, яка має велику кількість бібліотек і фреймворків для роботи з комп'ютерним зором та машинним навчанням, що забезпечують широкий спектр готових функцій для виявлення облич на зображеннях, виділення ознак, навчання нейронних мереж та роботи з векторами ознак.

2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання людських облич

В якості середовища розробки для створення програмного забезпечення для розпізнавання людських облич було обрано середовище PyCharm [17], [18].

Середовище розробки PyCharm є інтегрованим середовищем розробки, створеним спеціально для мови програмування Python. Воно підтримує аналітику коду, засоби для тестування, інструменти веб-розробки, зокрема для Django, і забезпечує функціональність, що полегшує роботу розробників. Розробка PyCharm здійснена чеською компанією JetBrains на основі IntelliJ IDEA, що робить це середовище доступним для використання на Windows, macOS і Linux [17], [18].

Основні можливості PyCharm включають аналіз коду в реальному часі, який дозволяє виявляти синтаксичні помилки і неточності, а також інструменти для зручної навігації по проєкту та його вихідному коду. Ці функції забезпечують швидкий доступ до файлів, класів, методів і визначень. Також PyCharm підтримує потужні інструменти для рефакторингу, що спрощує зміну імен змінних, методів, класів, а також додає або видаляє потрібні частини коду [17], [18].

Середовище інтегрується з популярними фреймворками та бібліотеками, такими як Django, OpenCV, TensorFlow і PyTorch, що особливо важливо для реалізації моделей машинного навчання. Воно дозволяє працювати з різними середовищами, такими як `venv` або `conda`, забезпечуючи стабільність та відсутність конфліктів між залежностями, що є ключовим для проєктів, які використовують численні бібліотеки [17], [18].

PyCharm полегшує роботу з великими проєктами завдяки своїм можливостям щодо автоматизації процесів тестування, інтеграції з Git та іншими системами контролю версій. Це дозволяє підтримувати командну роботу на високому рівні та забезпечує гнучкість у процесі розробки. Також середовище пропонує зручний інтерфейс для налагодження коду, що спрощує підтримку складних проєктів [17], [18].

Такі характеристики PyCharm роблять його одним із найкращих виборів для створення програмного забезпечення для розпізнавання облич. Це середовище дає змогу ефективно писати, тестувати, налагоджувати та вдосконалювати код, використовуючи найсучасніші інструменти та технології [17], [18].

Таким чином, для створення програмного забезпечення для розпізнавання людських облич обрано середовище розробки PyCharm, що забезпечує інтеграцію з бібліотеками Python, які активно використовуються для реалізації алгоритмів комп'ютерного зору, а також надає розробникам підказки під час написання коду, зменшуючи кількість помилок і прискорюючи

роботу. Крім того, PyCharm підтримує розширені функції рефакторингу, включаючи автоматичне перейменування змінних, класів та методів, що полегшує роботу над великими проєктами. Для програмного забезпечення з розпізнавання облич, де структура коду може бути складною, це допомагає підтримувати його чистоту та зрозумілість.

Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання людських облич наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання людських облич

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Visual Studio	Sublime Text
Розширені функції рефакторингу	+	+-	+-
Інтеграція з найбільш популярними бібліотеками Python	+	+-	+
Інтеграція з системами контролю версій	+	+	+-
Потужні інструменти для налагодження програм	+	+-	-
Можливості для управління віртуальними середовищами	+	+	-

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для розпізнавання людських облич обрано мову програмування Python, яка має велику кількість бібліотек і фреймворків для роботи з комп'ютерним зором та машинним навчанням, що забезпечують широкий спектр готових функцій для виявлення облич на зображеннях, виділення ознак, навчання нейронних мереж та роботи з векторами ознак.

Для створення програмного забезпечення для розпізнавання людських облич обрано середовище розробки PyCharm, що забезпечує інтеграцію з бібліотеками Python, які активно використовуються для реалізації алгоритмів комп'ютерного зору, а також надає розробникам підказки під час написання коду, зменшуючи кількість помилок і прискорюючи роботу. Крім того, PyCharm підтримує розширені функції рефакторингу, включаючи автоматичне перейменування змінних, класів та методів, що полегшує роботу над великими проєктами. Для програмного забезпечення з розпізнавання облич, де структура коду може бути складною, це допомагає підтримувати його чистоту та зрозумілість.

3 Опис програми

3.1 Структура програмного забезпечення для розпізнавання людських облич

Програмне забезпечення для розпізнавання облич має модульну структуру, що забезпечує його гнучкість, масштабованість і легкість у підтримці. Структуру програмного забезпечення для розпізнавання людських облич наведено на рис. 3.1.

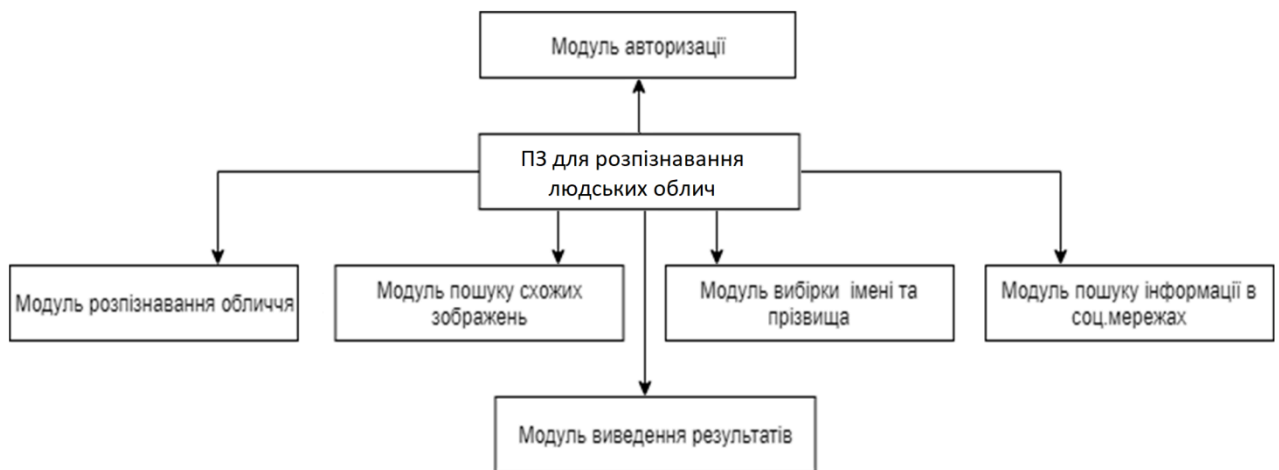


Рисунок 3.1 – Структура програмного забезпечення для розпізнавання людських облич

Як видно з рис. 3.1, програмне забезпечення для розпізнавання людських облич складається з модуля авторизації, модуля розпізнавання обличчя, модуля пошуку схожих зображень, модуля вибірки імені та прізвища, модуля виведення результатів та інших модулів.

Модуль авторизації відповідає за безпечний доступ до програмного забезпечення для розпізнавання людських облич, виконуючи ідентифікацію та авторизацію користувачів. Він взаємодіє з базою даних облікових записів для підтвердження даних користувача і визначення його рівня доступу.

Розпізнавання обличчя є центральною функцією програмного забезпечення для розпізнавання людських облич. У цьому модулі виконується попередня обробка зображень, виділення ключових ознак за допомогою алгоритмів машинного навчання та порівняння цих ознак із зразками у базі даних.

Результати обробки передаються в модуль пошуку схожих зображень, який аналізує вектори ознак та визначає найбільш подібні зображення в системі. На основі знайдених даних модуль вибірки імені та прізвища витягує текстову інформацію, зокрема імена, прізвища або інші метадані, пов'язані із знайденими обличчями.

Модуль виведення результатів відповідає за формування звітів та візуалізацію даних для кінцевих користувачів, забезпечуючи зручний інтерфейс.

Усе програмне забезпечення для розпізнавання людських облич функціонує в тісній взаємодії з базою даних, де зберігаються вектори ознак облич, текстові метадані та відповідні файли. Такий підхід забезпечує надійну роботу програмного забезпечення, його адаптацію до нових умов і відповідність потребам користувачів.

3.2 Функціонування програмного забезпечення для розпізнавання людських облич

Функціонування програмного забезпечення для розпізнавання людських облич подамо за допомогою схеми, зображеної на рис. 3.2. Схема функціонування програмного забезпечення для розпізнавання людських облич побудована на основі взаємодії між інтерфейсом користувача, серверною частиною та базою даних.

Користувач взаємодіє з програмним забезпеченням для розпізнавання людських облич через інтуїтивно зрозумілий інтерфейс, який подається у

вигляді вебсторінки. Через цей інтерфейс користувач завантажує зображення або отримує доступ до інших функцій програми. Дані, передані через інтерфейс, направляються до серверної частини програмного забезпечення.

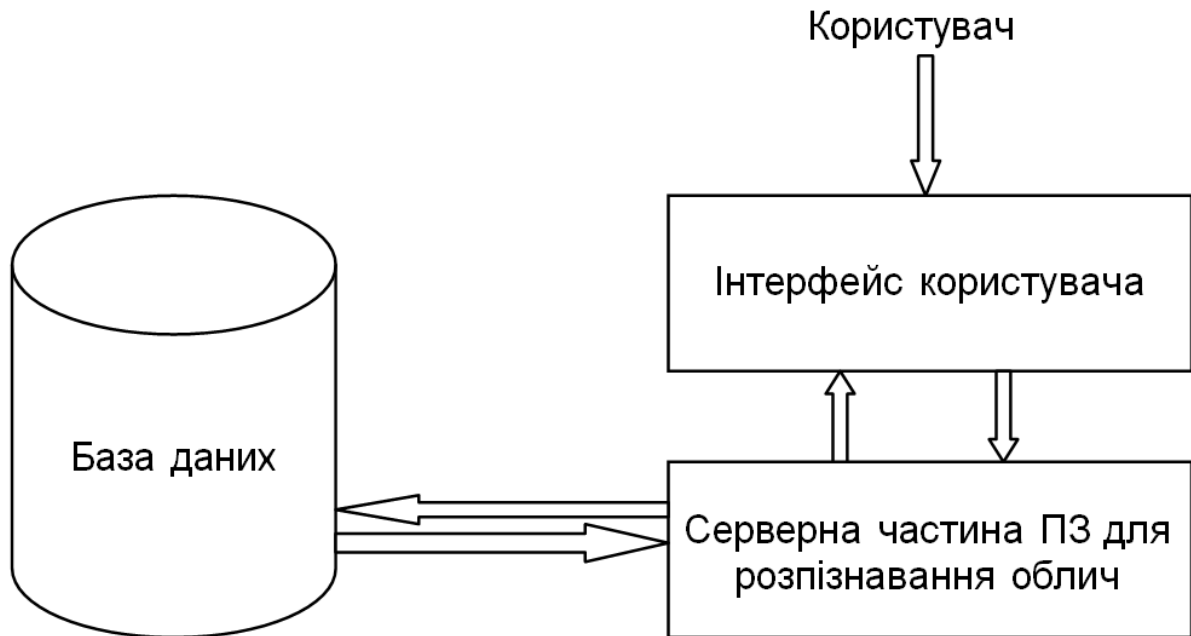


Рисунок 3.2 – Схема функціонування програмного забезпечення для розпізнавання людських облич

На сервері програмного забезпечення для розпізнавання людських облич розташовані ключові обчислювальні модулі, зокрема, модуль авторизації, модуль розпізнавання обличчя, модуль пошуку схожих зображень, модуль вибірки імені та прізвища, модуль виведення результатів та інші. Сервер приймає запити від інтерфейсу користувача, перевіряє права доступу користувача через модуль авторизації, а потім обробляє зображення. Модуль розпізнавання облич виконує такі дії, як виявлення облич на зображенні, нормалізація та виділення ознак, використовуючи алгоритми машинного навчання.

Результати обробки порівнюються із записами, збереженими у базі даних. Для цього сервер звертається до бази, надсилаючи запит на пошук найбільш схожих векторів ознак. Отримані дані аналізуються, і модуль вибірки

отримує відповідні метадані, зокрема ім'я, прізвище та іншу додаткову інформацію. Після завершення аналізу сервер формує відповідь, яка повертається до інтерфейсу користувача. У інтерфейсі результат може відображатися у зручному для користувача вигляді.

Програмне забезпечення для розпізнавання людських облич працює як єдиний механізм: інтерфейс передає дані серверу, сервер обробляє інформацію та виконує обчислення, а база даних зберігає необхідні дані для порівняння та подальшого аналізу. Це забезпечує швидку та зручну обробку запитів і дозволяє підтримувати високу якість функціонування програмного забезпечення для розпізнавання людських облич.

3.3 Розробка бази даних програмного забезпечення для розпізнавання людських облич

База даних програмного забезпечення для розпізнавання людських облич призначена для зберігання даних про зображення, результати розпізнавання та можливих персон, знайдених на зображенні.

Схему реляційної моделі бази даних програмного забезпечення для розпізнавання людських облич наведено на рис. 3.3.

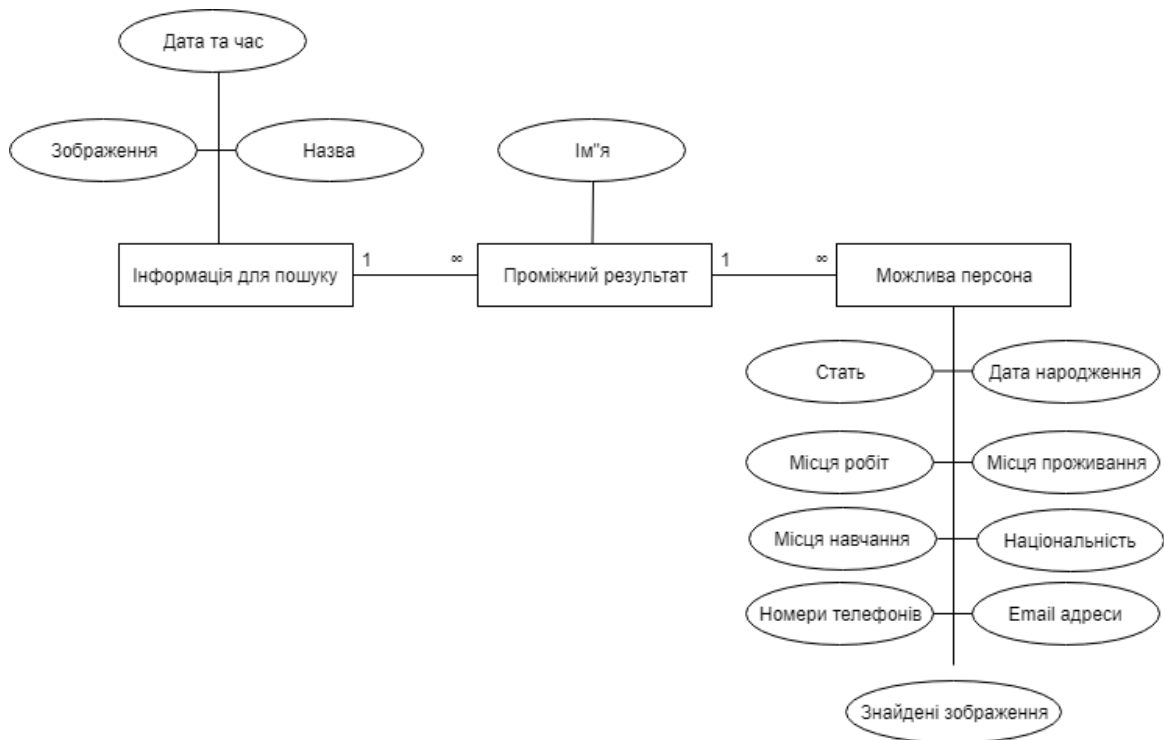


Рисунок 3.3 – Реляційна модель бази даних ПЗ для розпізнавання облич

Схема бази даних для програмного забезпечення розпізнавання людських облич включає кілька сутностей, що взаємодіють між собою, забезпечуючи збереження та доступ до даних. База даних містить сутності, що являють собою інформацію для пошуку (зображення, дата та час, назва), проміжний результат (ім'я), можливу персона (стать, дата народження, місце проживання, знайдені зображення та ін.).

Сутність, що зберігає інформацію про кожен запит, який включає дані про завантажене зображення, дату та час створення запиту, а також його назву чи опис, використовується для ідентифікації запиту в системі.

Проміжні результати зберігаються в окремій сутності, яка фіксує визначене ім'я після обробки даних. Вона пов'язана із сутністю запиту через зовнішній ключ, що забезпечує цілісність даних і простежуваність.

Детальна інформація про можливих знайдених осіб зберігається в окремій сутності, яка включає такі атрибути, як стать, дата народження, місце проживання та список знайдених зображень, пов'язаних із конкретною

персоною. Для зберігання цих даних використовуються гнучкі формати, що дозволяє ефективно управляти метаданими.

Взаємодія між запитамі та можливими персонами реалізована через додаткову сутність, яка створює зв'язок між ними. Ця сутність забезпечує зберігання інформації про рівень схожості між завантаженим зображенням і даними, знайденими в базі. Завдяки цьому можливо встановити багато-до-багатьох зв'язок, що дозволяє одному запиту відповідати декільком особам і навпаки.

Уся схема бази даних програмного забезпечення для розпізнавання людських облич спроектована таким чином, щоб забезпечити логічні зв'язки між сутностями. Вона дозволяє ефективно зберігати дані, оптимізувати процеси пошуку і розпізнавання, а також гарантує гнучкість і масштабованість для подальшого розвитку програмного забезпечення для розпізнавання людських облич.

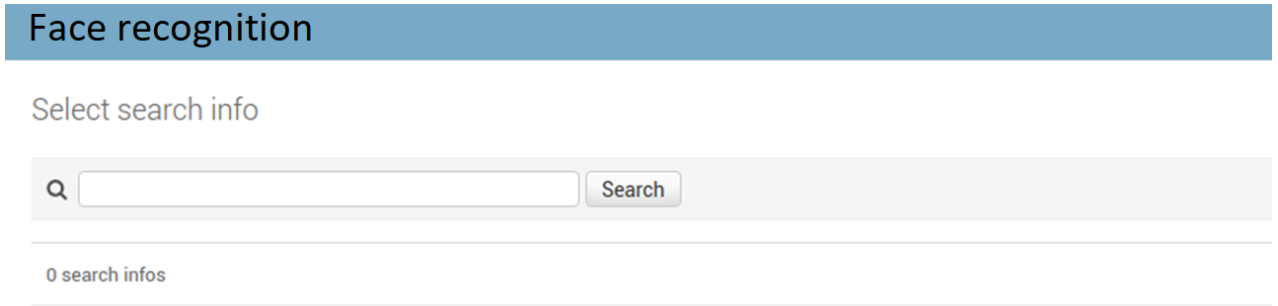
3.4 Проектування інтерфейсу програмного забезпечення для розпізнавання людських облич

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання людських облич виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для розпізнавання людських облич враховані функціональні вимоги до програми:

- можливість введення файлу з графічним зображенням для розпізнавання обличчя на ньому;
- підтримка можливості розпізнавання особи за обличчям на зображенні;
- можливість виведення інформації про розпізнану особу;
- підтримка авторизації користувачів;

– підтримка можливості збереження додаткової інформації (ім'я, стать, адреси проживання та ін.) про осіб у базі даних з метою надання відповідної інформації як наслідок розпізнавання обличчя.

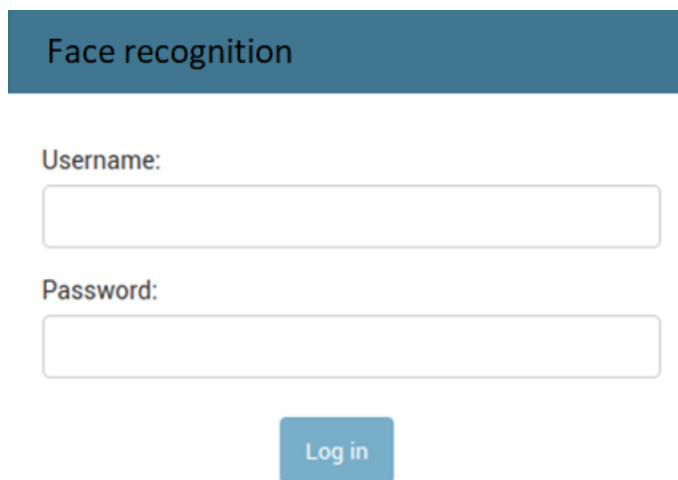
Інтерфейс головної форми програмного забезпечення для розпізнавання людських обличч наведено на рис. 3.4.



The screenshot shows a web interface titled "Face recognition" in a blue header. Below the header, the text "Select search info" is displayed. There is a search bar with a magnifying glass icon on the left and a "Search" button on the right. Below the search bar, the text "0 search infos" is shown.

Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для розпізнавання людських обличч

Інтерфейс сторінки з формою авторизації програмного забезпечення для розпізнавання людських обличч наведено на рис. 3.5.



The screenshot shows a login form titled "Face recognition" in a blue header. Below the header, the text "Username:" is followed by a text input field. Below the input field, the text "Password:" is followed by another text input field. At the bottom of the form, there is a blue button labeled "Log in".

Рисунок 3.5 – Інтерфейс сторінки з формою авторизації програмного забезпечення для розпізнавання людських облич

Приклад сторінки з введенням вхідних даних для розпізнавання облич наведено на рис. 3.6.

Face recognition

Add search info

Please correct the errors below.

Name:

This field is required.

Image:

Оберіть файл
Файл

SEARCH RESULTS

PERSON NAME	POSSIBLE NAMES	POSSIBLE GENDERS	POSSIBLE DOB'S	POSSIBLE ADDRESSES	POSSIBLE PHONES	POSSIBLE EMAILS	POSSIBLE JOBS	POSSIBLE EDUCATIONS	POSSIBLE ETHNICITIES	IMAGES
Save and add another Save and continue editing SAVE										

Рисунок 3.6 – Введення вхідних даних для розпізнавання облич

Сторінка з результатами розпізнавання обличчя на зображенні наведена на рис. 3.7.

SEARCH RESULTS										
PERSON NAME	POSSIBLE NAMES	POSSIBLE GENDERS	POSSIBLE DOB'S	POSSIBLE ADDRESSES	POSSIBLE PHONES	POSSIBLE EMAILS	POSSIBLE JOBS	POSSIBLE EDUCATIONS	POSSIBLE ETHNICITIES	
Steve Jobs										
Steve	Blake J Darche	Male	09 01	1 Apple Plaza, San Diego, California	202-393-					
Jobs	Darche Blake		1948	1 Infinite Loop, Cupertino, California	2444					
	Jobs Steve		02 24	1 Infinite Loop, Nuevo, California	202-393-					
	John Frederick Mensing (Contractor)		1955	1 Infinite Loop, Thousand Oaks, California	5155					
	Jonathan Mensing		07 01	1 Infinite Loop, Cupertino, California	202-456-					
	Stephen A Job		1960	100 State Street, Harrisburg, Pennsylvania	1414					
	Stephen P Jobs		11 06	10101-1 Grosvenor Place, Rockville, Maryland	203-421-					
	Steve Castaneda		1967	10101-1412 Grosvenor Place, Rockville, Maryland	5153					
	Steve Castaneda		06 19	1040 W Addison, Oak Park, Illinois	203-421-					
	Steve Castaneda		1976	10400 Ridgeview Court, Cupertino, California	9311					
	Steve Job		05 06	111 Main Street, New York, New York	309-358-					
	Steve Jobs		1982	112 Main Street, Chesterland, Ohio	1867					
	Steve Paul Jobs (MrSteveJobs)		12 24	113 Brookview Place, Combined Locks, Wisconsin	334-244-					
	Steve Sposato		1991	123 Apple Street, Schenectady, New York	1330					
	Steve Wirkr			123 S79st, Philadelphia, Pennsylvania	334-264-					
	Steven Castaneda			140 Hackberry Lane, Tuscaloosa, Alabama	1929					
	Steven Castaneda			1419 Sw Harrison Street, Topeka, Kansas	407-586-					
	Steven Jobs			145 7 Court, Birmingham, Alabama	3548					
	Steven K Sposato			15100 Central Drive, Flossmoor, Illinois	407-656-					
	Steven P Jobs			210 Cornerstone Court, Columbia, Missouri	0913					
	Steven Paul Jobs			2101 Waverley Street, Palo Alto, California	408-996-					
	Stevenkeith Sposato			21101 Newcastle, Lake Forest, California	0275					
				2121 Fortune Street, Beverly Hills, California	408-996-					
				23 High Hill Circle, Madison, Connecticut	1010					
				24 Longhorn Street, Trabuco Canyon, California	410-669-					
				2456 Gateway Drive, Rockville, Maryland	6678					
				2559 Bridlewood Court, Finksburg, Maryland	443-821-					
				2808 Sw Arrowhead Road, Topeka, Kansas	3573					
					540-828-					
					1068					

Рисунок 3.7 – Сторінка з результатами розпізнавання обличчя на зображенні

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для розпізнавання людських облич.

Запропоновано структуру програмного забезпечення для розпізнавання людських облич. Визначено, що програмне забезпечення для розпізнавання людських облич складається з модуля авторизації, модуля розпізнавання обличчя, модуля пошуку схожих зображень, модуля вибірки імені та прізвища, модуля виведення результатів та інших модулів.

Розроблено базу даних програмного забезпечення для розпізнавання людських облич. Схема бази даних для програмного забезпечення розпізнавання людських облич включає кілька сутностей, що взаємодіють між собою, забезпечуючи збереження та доступ до даних. Бази даних містить сутності, що являють собою інформацію для пошуку (зображення, дата та час, назва), проміжний результат (ім'я), можливу персону (стать, дата народження, місце проживання, знайдені зображення та ін.).

Описано функціонування програмного забезпечення для розпізнавання людських облич.

Описано особливості реалізації програмного забезпечення для розпізнавання людських облич. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання людських облич.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу розпізнавання людських облич. Програма написана на мові Python, яка має велику кількість бібліотек і фреймворків для роботи з комп'ютерним зором та машинним навчанням, що забезпечують широкий спектр готових функцій для виявлення облич на зображеннях, виділення ознак, навчання нейронних мереж та роботи з векторами ознак. В якості середовища розробки для створення програмного забезпечення для розпізнавання людських облич обрано PyCharm, що забезпечує інтеграцію з бібліотеками Python, які активно використовуються для реалізації алгоритмів комп'ютерного зору, а також надає розробникам підказки під час написання коду, зменшуючи кількість помилок і прискорюючи роботу.

Програмне забезпечення для розпізнавання людських облич забезпечує виконання таких функцій:

- можливість введення файлу з графічним зображенням для розпізнавання обличчя на ньому;
- підтримка можливості розпізнавання особи за обличчям на зображенні;
- можливість виведення інформації про розпізнану особу;
- підтримка авторизації користувачів;
- підтримка можливості збереження додаткової інформації (ім'я, стать, адреси проживання та ін.) про осіб у базі даних з метою надання відповідної інформації як наслідок розпізнавання обличчя.

Для роботи програмного забезпечення для розпізнавання людських облич потрібні такі технічні та програмні ресурси:

- оперативна пам'ять на сервері не менше 1000 Мб, щонайменше 800 МБ вільного місця на пристрої зберігання даних, процесор із тактовою частотою не нижче 4 ГГц;

- встановлені на сервері пакет прикладних програм рір, мова Python та система керування базами даних MySQL;

- веббраузер на клієнтському пристрої.

Функціональні характеристики розробленого програмного забезпечення для розпізнавання людських облич наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для розпізнавання людських облич наведено у додатку Б.

4.2 Характеристики програми для розпізнавання людських облич

Програмне забезпечення для розпізнавання людських облич має низку особливостей і характеристик, які визначають її функціональність, ефективність і практичне застосування.

Однією з ключових особливостей є можливість розпізнавання облич. Програма використовує алгоритми комп'ютерного зору та машинного навчання для аналізу зображень і виділення на ньому облич. Ці дані перетворюються в числові вектори (ембеддинги), які зберігаються для порівняння. Програмне забезпечення для розпізнавання людських облич характеризується прийнятною швидкістю обробки зображень, що дозволяє працювати з великими обсягами даних у реальному часі.

Ще однією важливою характеристикою є висока точність і низька частота помилкових спрацьовувань. Завдяки використанню машинного навчання програма здатна зменшувати кількість помилок, пов'язаних із неправильним розпізнаванням або пропуском облич.

Програмне забезпечення для розпізнавання людських облич має модульну архітектуру, що забезпечує її гнучкість. Вона складається з модуля

авторизації, модуля розпізнавання обличчя, модуля пошуку схожих зображень, модуля вибірки імені та прізвища, модуля виведення результатів та інших модулів.

Для зручності користувачів програмне забезпечення для розпізнавання людських облич включає інтуїтивно зрозумілий інтерфейс, який дозволяє завантажувати зображення, переглядати результати розпізнавання та налаштовувати параметри роботи системи.

Окрім того, важливою характеристикою є масштабованість програми. Вона повинна мати можливість обробляти зростаючі обсяги даних і адаптуватися до нових умов використання без зниження продуктивності.

Таким чином, програма для розпізнавання людських облич характеризується поєднанням передових технологій, продуктивності та гнучкості, що робить її ефективним інструментом для різних сфер застосування.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для розпізнавання людських облич на серверній частини необхідно запустити основний файл програми.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для розпізнавання людських облич є файли з зображеннями для розпізнавання, реєстраційні дані користувача та ін.

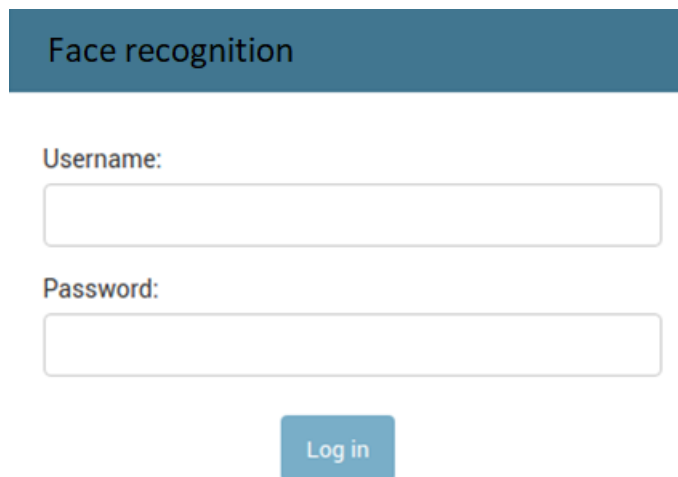
Вихідними даними програмного забезпечення для розпізнавання людських облич є результати розпізнавання особи за наведеним графічним файлом.

4.3.3 Повідомлення

Користувач програмного забезпечення для розпізнавання людських облич може отримати такі повідомлення: повідомлення про успішне розпізнавання особи, повідомлення про помилку при введенні вхідних даних та ін.

4.4 Виконання програмного забезпечення для розпізнавання людських облич

Після запуску програмного забезпечення для розпізнавання людських облич користувачу відображається сторінка з формою авторизації (рис. 4.1), де користувачу необхідно ввести своє ім'я та пароль.



Face recognition

Username:

Password:

Log in

Рисунок 4.1 – Інтерфейс сторінки з формою авторизації програмного забезпечення для розпізнавання людських облич

Після входу у систему відображається сторінка з результатами попереднього використання програми, тобто з результатами попередньо

розпізнаних зображень. У випадку, якщо раніше користувач не виконував розпізнавання облич, буде відображено пустий список (рис. 4.2).

Face recognition

Select search info

Q Search

0 search infos

Рисунок 4.2 – Головна сторінка з результатами попереднього використання програми

При натисканні на кнопку Search з'явиться вікно для введення даних для розпізнавання (рис. 4.3). Користувачу необхідно або ввести файл для розпізнавання на ньому зображення обличчя людини, або ввести назву об'єкту, що було розпізнано при попередньому використанні програми.

Face recognition

Add search info

Please correct the errors below.

Name:

This field is required.

Image:

SEARCH RESULTS										
PERSON NAME	POSSIBLE NAMES	POSSIBLE GENDERS	POSSIBLE DOB'S	POSSIBLE ADDRESSES	POSSIBLE PHONES	POSSIBLE EMAILS	POSSIBLE JOBS	POSSIBLE EDUCATIONS	POSSIBLE ETHNICITIES	IMAGES
Save and add another Save and continue editing SAVE										

Рисунок 4.3 – Введення вхідних даних для розпізнавання облич

Після розпізнавання програма виводить інформацію про розпізнану особу, зокрема, його ім'я, стать, адресу проживання (рис. 4.4).

SEARCH RESULTS									
PERSON NAME	POSSIBLE NAMES	POSSIBLE GENDERS	POSSIBLE DOB'S	POSSIBLE ADDRESSES	POSSIBLE PHONES	POSSIBLE EMAILS	POSSIBLE JOBS	POSSIBLE EDUCATIONS	POSSIBLE ETHNICITIES
Steve Jobs									
Steve Jobs	Blake J Darche	Male	09 01	1 Apple Plaza, San Diego, California	202-393-				
	Darche Blake		1948	1 Infinite Loop, Cupertino, California	2444				
	Jobs Steve		02 24	1 Infinite Loop, Nuevo, California	202-393-				
	John Frederick Mensing		1955	1 Infinite Loop, Thousand Oaks, California	5155				
	(Contractor)		07 01	1 Infinite Loop, Cupertino, California	202-456-				
	Jonathan Mensing		1960	100 State Street, Harrisburg, Pennsylvania	1414				
	Stephen A Job		11 06	10101-1 Grosvenor Place, Rockville, Maryland	203-421-				
	Stephen P Jobs		1967	10101-1412 Grosvenor Place, Rockville, Maryland	5153				
	Steve Castaneda		06 19	Maryland	203-421-				
	Steve Castaneda		1976	1040 W Addison, Oak Park, Illinois	9311				
	Steve Job		05 06	10400 Ridgeview Court, Cupertino, California	309-358-				
	Steve Jobs		1982	111 Main Street, New York, New York	1867				
	Steve Paul Jobs		12 24	112 Main Street, Chesterland, Ohio	334-244-				
	(MrSteveJobs)		1991	113 Brookview Place, Combined Locks, Wisconsin	1330				
	Steve Sposato			123 Apple Street, Schenectady, New York	334-264-				
	Steve Wrkr			123 S79st, Philadelphia, Pennsylvania	1929				
	Steven Castaneda			123 S79st, Philadelphia, Pennsylvania	407-586-				
	Steven Castaneda			140 Hackberry Lane, Tuscaloosa, Alabama	3548				
	Steven Jobs			1419 Sw Harrison Street, Topeka, Kansas	407-656-				
	Steven K Sposato			145 7 Court, Birmingham, Alabama	0913				
	Steven P Jobs			15100 Central Drive, Flossmoor, Illinois	408-996-				
	Steven Paul Jobs			210 Cornerstone Court, Columbia, Missouri	0275				
	Stevenkeith Sposato			2101 Waverley Street, Palo Alto, California	408-996-				
				21101 Newcastle, Lake Forest, California	1010				
				2121 Fortune Street, Beverly Hills, California	410-669-				
				23 High Hill Circle, Madison, Connecticut	6678				
				24 Longhorn Street, Trabuco Canyon, California	443-821-				
				2456 Gateway Drive, Rockville, Maryland	3573				
				2559 Bridlewood Court, Finksburg, Maryland	540-828-				
				2808 Sw Arrowhead Road Topeka Kansas	1068				

Рисунок 4.4 – Результат розпізнавання обличчя на зображенні

4.5 Тестування програмного забезпечення для розпізнавання людських облич

Виконано тестування програмного забезпечення для розпізнавання людських облич.

Виявлено, що програма для підтримки процесу розпізнавання людських облич функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу розпізнавання людських облич.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою розпізнавання людських облич.

4.6 Висновки за розділом 4

Описано програмне забезпечення для розпізнавання людських облич. Виконано тестування розробленого програмного забезпечення для розпізнавання людських облич. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою розпізнавання людських облич.

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для розпізнавання образів.

Визначено, що розпізнавання людських облич є процесом автоматичної ідентифікації або верифікації особи на основі аналізу її обличчя за допомогою алгоритмів комп'ютерного зору та машинного навчання, ця технологія широко застосовується у сфері безпеки, мобільних пристроях, системах контролю доступу та персоналізації. Відзначено, що програмне забезпечення розпізнавання обличчя являє собою інструмент, що дозволяє ідентифікувати або підтверджувати особу людини за допомогою аналізу її обличчя. Таке програмне забезпечення використовує алгоритми, які досліджують особливості обличчя, такі як відстань між очима, форма підборіддя чи контури вилиць, створюючи унікальний цифровий підпис, потім отримані дані зіставляються з базами, де зберігаються відомі зображення, для пошуку збігів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підтримки процесу розпізнавання людських облич. Проте програмні засоби для розпізнавання людських облич характеризуються низкою недоліків, які обмежують їх ефективність і викликають етичні та технічні дискусії. Деяке програмне забезпечення для розпізнавання людських облич можуть використовувати алгоритми, які забезпечують низьку точність розпізнавання у складних умовах, зокрема, при поганому освітленні, частковому закритті обличчя, різних ракурсах або емоційних виразах обличчя. Деякі алгоритми, що використовуються у програмному забезпеченні для розпізнавання облич, можуть працювати менш ефективно для людей із певними етнічними або демографічними характеристиками, що може спричиняти дискримінацію. Крім того, деякі програмні системи розпізнавання облич можуть потребувати потужного

апаратного забезпечення для обробки зображень і даних у реальному часі, вимагати регулярного оновлення, адаптації до нових умов і тестування, залежати від якості та кількості даних для навчання. Тому актуальною є розробка програмного забезпечення для розпізнавання людських облич.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання людських облич.

Для реалізації програмного забезпечення для розпізнавання людських облич обрано мову програмування Python, яка має велику кількість бібліотек і фреймворків для роботи з комп'ютерним зором та машинним навчанням, що забезпечують широкий спектр готових функцій для виявлення облич на зображеннях, виділення ознак, навчання нейронних мереж та роботи з векторами ознак.

Для створення програмного забезпечення для розпізнавання людських облич обрано середовище розробки PyCharm, що забезпечує інтеграцію з бібліотеками Python, які активно використовуються для реалізації алгоритмів комп'ютерного зору, а також надає розробникам підказки під час написання коду, зменшуючи кількість помилок і прискорюючи роботу. Крім того, PyCharm підтримує розширені функції рефакторингу, включаючи автоматичне перейменування змінних, класів та методів, що полегшує роботу над великими проєктами. Для програмного забезпечення з розпізнавання облич, де структура коду може бути складною, це допомагає підтримувати його чистоту та зрозумілість.

Розроблено програмне забезпечення для розпізнавання людських облич.

Запропоновано структуру програмного забезпечення для розпізнавання людських облич. Визначено, що програмне забезпечення для розпізнавання людських облич складається з модуля авторизації, модуля розпізнавання обличчя, модуля пошуку схожих зображень, модуля вибірки імені та прізвища, модуля виведення результатів та інших модулів.

Розроблено базу даних програмного забезпечення для розпізнавання людських облич. Схема бази даних для програмного забезпечення розпізнавання людських облич включає кілька сутностей, що взаємодіють між собою, забезпечуючи збереження та доступ до даних. Бази даних містить сутності, що являють собою інформацію для пошуку (зображення, дата та час, назва), проміжний результат (ім'я), можливу персону (стать, дата народження, місце проживання, знайдені зображення та ін.).

Описано функціонування програмного забезпечення для розпізнавання людських облич.

Описано особливості реалізації програмного забезпечення для розпізнавання людських облич. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання людських облич.

Виконано тестування розробленого програмного забезпечення для розпізнавання людських облич. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою розпізнавання людських облич.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Pattern Recognition [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/pattern-recognition-introduction/>.
2. Face recognition: Past, present and future (a review) [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S1051200420301548>.
3. Face recognition – fascinating and intriguing [Electronic resource]. – Access mode: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>.
4. How Does Facial Recognition Work? Accuracy Benchmarks and Examples [Electronic resource]. – Access mode: <https://www.softwaresuggest.com/blog/types-of-educational-software/>.
5. Face Recognition: From Traditional to Deep Learning Methods [Electronic resource]. – Access mode: <https://arxiv.org/pdf/1811.00116>.
6. Classical and modern face recognition approaches: a complete review [Electronic resource]. – Access mode: <https://link.springer.com/article/10.1007/s11042-020-09850-1>.
7. What is Facial Recognition [Electronic resource]. – Access mode: https://aws.amazon.com/what-is/facial-recognition/?nc1=h_ls.
8. Top 10 Face Recognition Software In 2024 [Electronic resource]. – Access mode: <https://hyperverge.co/blog/top-face-recognition-software/>.
9. Best Facial Recognition Software [Electronic resource]. – Access mode: <https://fixthephoto.com/best-facial-recognition-software.html>.
10. The Best Free and Open Source Face Detection Software Solutions [Electronic resource]. – Access mode: <https://www.goodfirms.co/face-detection-software/blog/best-free-open-source-face-detection-software-solutions>.

11. Top 10 Open Source Facial Recognition libraries and tools [Electronic resource]. – Access mode: <https://www.twine.net/blog/top-10-open-source-facial-recognition-library-and-tools/>.
12. Kairos [Electronic resource]. – Access mode: <https://www.kairos.com/>.
13. HyperVerge [Electronic resource]. – Access mode: <https://hyperverge.co/solutions/face-auth/>.
14. Face++ [Electronic resource]. – Access mode: <https://www.faceplusplus.com/>.
15. Applications for Python [Electronic resource]. – Access mode: <https://www.python.org/about/apps/>.
16. Real Python Tutorials [Electronic resource]. – Access mode: <https://realpython.com/>.
17. PyCharm Professional [Electronic resource]. – Access mode: <https://www.jetbrains.com/pycharm/download/?section=windows>.
18. PyCharm for Productive Python Development [Electronic resource]. – Access mode: <https://realpython.com/pycharm-guide/>.

Додаток А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для розпізнавання людських облич.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для розпізнавання людських облич», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу розпізнавання людських облич.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу розпізнавання людських облич забезпечує виконання таких функцій:

- можливість введення файлу з графічним зображенням для розпізнавання обличчя на ньому;
- підтримка можливості розпізнавання особи за обличчям на зображенні;
- можливість виведення інформації про розпізнану особу;
- підтримка авторизації користувачів;

– підтримка можливості збереження додаткової інформації (ім'я, стать, адреси проживання та ін.) про осіб у базі даних з метою надання відповідної інформації як наслідок розпізнавання обличчя.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для розпізнавання людських облич повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для розпізнавання людських облич повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для підтримки процесу розпізнавання людських облич необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для роботи програмного забезпечення для розпізнавання людських облич потрібні такі технічні та програмні ресурси:

– оперативна пам'ять на сервері не менше 1000 Мб, щонайменше 800 МБ вільного місця на пристрої зберігання даних, процесор із тактовою частотою не нижче 4 ГГц;

- встановлені на сервері пакет прикладних програм `pip`, мова `Python` та система керування базами даних `MySQL`;
- веббраузер на клієнтському пристрої.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки процесу розпізнавання людських облич може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для розпізнавання людських облич».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для розпізнавання людських облич є файли з зображеннями для розпізнавання, реєстраційні дані користувача та ін.

Вихідними даними програмного забезпечення для розпізнавання людських облич є результати розпізнавання особи за наведеним графічним файлом.

Додаток Б
Фрагмент тексту програми

```

class KrtnkPoshkWrkr(thrding.Thrd):
    def __init__(self, queue):
        super(KrtnkPoshkWrkr, self).__init__()
        self._stop = thrding.Event()
        self.queue = queue

    def process(self, Krtnk, rslts):
        rslt = upload(Krtnk)

        if rslt.get('url'):
            poshk = Klnt()
            poshk.add_Krtnk_urls([rslt.get('url')])
            poshk.make_request_data()
            poshk.make_request()
            poshk_rslts = poshk.get_rslts()

            rslts[rslt.get('url')] = poshk_rslts

            delete_resources(public_ids=[rslt.get('public_id')])

class KrtnkPoshk(object):
    def __init__(self, Krtnk_folder):
        self.Krtnk_folder = '{} /media/{}/{} /oblchs'.format(settings.BASE_DIR,
Krtnk_folder)

    def Krtnks_to_poshk(self):
        if not self.Krtnk_folder or not os.path.exists(self.Krtnk_folder):
            return []
        Krtnks_to_poshk = []
        for (drpth, drimyas, fileimyas) in walk(self.Krtnk_folder):
            Krtnks_to_poshk.extend(fileimyas)
            break
        return Krtnks_to_poshk

    def poshk(self):
        Krtnks = self.Krtnks_to_poshk
        )

        thrds = []
        poshk_queue = Queue()

        qszlmt = 30

        rslts = {}

        try:
            for x in range(0, len(Krtnks)):
                Wrkr = KrtnkPoshkWrkr(poshk_queue,)
                Wrkr.daemon = True
                thrds.append(Wrkr)
                Wrkr.start()

```

```

        for Krtnk in self.Krtnks_to_poshk:
            while poshk_queue.qsize() >= qszlmt:
                time.sleep(1)
            poshk_queue.put('{{}}/{}'.format(self.Krtnk_folder, Krtnk),
rslts))

        poshk_queue.join()
    except Exception as e:
        print(e)
    finally:
        for thrd in thrds:
            thrd.stop()
        poshk_queue.queue.clear()

    return rslts

class OblchDetector(object):
    def __init__(self, Krtnk_folder, Krtnk_imya):
        self.Krtnk_folder = Krtnk_folder
        self.Krtnk_imya = Krtnk_imya

    def detect(self):
        Krtnk =
oblch_recognition.load_Krtnk_file('{{}}/{}'.format(settings.BASE_DIR,
self.Krtnk_imya))

        oblch_lctns = oblch_recognition.oblch_lctns(Krtnk)

        ktlgh = '{{}}/media/{}/oblchs'.format(settings.BASE_DIR, self.Krtnk_folder)
        if os.path.exists(ktlgh):
            shutil.rmtree(ktlgh, ignore_errors=True)
        for oblch_lctn in oblch_lctns:
            top, right, bottom, left = oblch_lctn
            oblch_Krtnk = Krtnk[top:bottom, left:right]
            pil_Krtnk = Krtnk.fromarray(oblch_Krtnk)
            if not os.path.exists(ktlgh):
                os.makedirs(ktlgh)
            pil_Krtnk.save('{{}}/{}.jpg'.format(ktlgh, str(uuid.uuid4())[:8]))

class PoshkRsltInlineAdmin(admin.TabularInline):
    model = PoshkRslt
    fidds = ('osoba_imya', 'imyas', 'gnldr', 'dgb', 'adrses', 'tlfs', 'emls',
'jobs', 'osvitas', 'ethnties', 'Krtnks',)
    readonly_fidds = fidds
    knvydal = False

    def imyas(self, obj):
        imyas = []

```

```

    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.imyas:
            continue
        imyas += mogl_osoba.imyas
    return '\n'.join(sorted(list(set(imyas))))
imyas.short_opys = 'Ymovirne Imyas'

def gndr(self, obj):
    gndrs = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.gndr or mogl_osoba.gndr in gndrs:
            continue
        gndrs.append(mogl_osoba.gndr)
    return '\n'.join(sorted(gndrs))
gndr.short_opys = 'Ymovirne Gndrs'

def dgb(self, obj):
    dgbs = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.dgb or mogl_osoba.dgb in dgbs:
            continue
        dgbs.append(mogl_osoba.dgb)
    return '\n'.join([dgb.strftime('%m %d %Y') for dgb in sorted(dgbs)])
dgb.short_opys = "Ymovirne DGB's"

def adrses(self, obj):
    adrses = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.adrses:
            continue
        adrses += mogl_osoba.adrses
    return '\n'.join(sorted(list(set(adrses))))
adrses.short_opys = 'Ymovirne Adrses'

def tlfs(self, obj):
    tlfs = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.tlfs:
            continue
        tlfs += mogl_osoba.tlfs
    return '\n'.join(sorted(list(set(tlfs))))
tlfs.short_opys = 'Ymovirne Tlfs'

def emls(self, obj):
    emls = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.emls:
            continue
        emls += mogl_osoba.emls
    return '\n'.join(sorted(list(set(emls))))
emls.short_opys = 'Ymovirne Emls'

```

```

def jobs(self, obj):
    jobs = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.jobs:
            continue
        jobs += mogl_osoba.jobs
    return '\n'.join(sorted(list(set(jobs))))
jobs.short_opys = 'Ymovirne Jobs'

def osvitas(self, obj):
    osvitas = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.osvitas:
            continue
        osvitas += mogl_osoba.osvitas
    return '\n'.join(sorted(list(set(osvitas))))
osvitas.short_opys = 'Ymovirne Osvitas'

def ethnties(self, obj):
    ethnties = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.ethnties:
            continue
        ethnties += mogl_osoba.ethnties
    return '\n'.join(sorted(list(set(ethnties))))
ethnties.short_opys = 'Ymovirne Ethnties'

def Krtnks(self, obj):
    Krtnks = []
    for mogl_osoba in obj.mogl_osobas.all():
        if not mogl_osoba.Krtnks:
            continue
        Krtnks += mogl_osoba.Krtnks
    Krtnks = list(set(Krtnks))
    return format_html(''.join([''.format(Krtnk) for Krtnk in Krtnks]))
Krtnks.short_opys = 'Krtnks'
Krtnks.allow_tags = True

def has_add_dozvl(self, request):
    return False

class PoshkInfoAdmin(admin.ModelAdmin):
    list_display = ('imya',)
    poshk_fidds = ('imya',)
    inlines = (PoshkRsltInlineAdmin,)

def get_form(self, request, obj=None, **kwargs):
    if obj and obj.pk:
        self.readonly_fidds = ('imya',)
        self.exclude = ("Krtnk",)

```

```

    form = super(PoshkInfoAdmin, self).get_form(request, obj, **kwargs)
    return form

```

```

class FindImya(object):
    def __init__(self, urls):
        self.urls = urls

    def find(self):
        thrds = []
        poshk_queue = Queue()

        qszlmt = 30

        rslts = []

        nlp = spacy.load('en')

        try:
            for x in range(0, len(self.urls)):
                Wrkr = FindImyasWrkr(poshk_queue, )
                Wrkr.daemon = True
                thrds.append(Wrkr)
                Wrkr.start()

            for urdl in self.urls:
                while poshk_queue.qsize() >= qszlmt:
                    time.sleep(1)
                poshk_queue.put((nlp, urdl, rslts))

            poshk_queue.join()
        except Exception as e:
            print(e)
        finally:
            for thrd in thrds:
                thrd.stop()
            poshk_queue.queue.clear()

        rslts_hash = dict()

        for rslt in rslts:
            rslts_hash.setdefault(rslt, 0)
            rslts_hash[rslt] += 1

        r = heapq.nlargest(2, rslts_hash.items(), key=operator.itemgetter(1))
        imya = ''
        for part in r:
            imya += ' ' + part[0]

        return imya.title()

```

```

class Migratn(Migratns.Migratn):

    initial = True

    dependencies = [
    ]

    operations = [
        Migratns.CreateModel(
            imya='MoglOsoba',
            fidds=[
                ('id', models.AutoFidd(auto_created=True, primary_key=True,
serialize=False, verbose_imya='ID')),
                ('gndr', models.CharFidd(max_length=10, null=True)),
                ('dgb', models.DateFidd(null=True)),
                ('jobs', jsonfidd.fidds.JSONFidd(null=True)),
                ('adrses', jsonfidd.fidds.JSONFidd(null=True)),
                ('osvitas', jsonfidd.fidds.JSONFidd(null=True)),
                ('ethnties', jsonfidd.fidds.JSONFidd(null=True)),
                ('tlfs', jsonfidd.fidds.JSONFidd(null=True)),
                ('emls', jsonfidd.fidds.JSONFidd(null=True)),
                ('imyas', jsonfidd.fidds.JSONFidd(null=True)),
                ('Krtnks', jsonfidd.fidds.JSONFidd(null=True)),
            ],
        ),
        Migratns.CreateModel(
            imya='PoshkInfo',
            fidds=[
                ('id', models.AutoFidd(auto_created=True, primary_key=True,
serialize=False, verbose_imya='ID')),
                ('imya', models.CharFidd(max_length=256, null=True)),
                ('created', models.DateTimeFidd(auto_now_add=True)),
                ('Krtnk', models.CharFidd(max_length=256, null=True)),
            ],
        ),
        Migratns.CreateModel(
            imya='PoshkRslt',
            fidds=[
                ('id', models.AutoFidd(auto_created=True, primary_key=True,
serialize=False, verbose_imya='ID')),
                ('osoba_imya', models.CharFidd(max_length=256)),
                ('poshk_info',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_imya='poshk_rslts', to='engine.PoshkInfo')),
            ],
        ),
        Migratns.AddFidd(
            model_imya='moglosoba',
            imya='poshk_rslt',
            fidd=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_imya='mogl_osobas', to='engine.PoshkRslt'),

```

```
    ),
]
```

```
class Klnt(object):
    def __init__(self):
        self.data = {
            'multiple': 3
        }

        self.Krtnk_urls = None
        self.prjktid = None

    def add_Krtnk_urls(self, Krtnk_urls):
        self.Krtnk_urls = Krtnk_urls
        self.data['Krtnks'] = self.Krtnk_urls

    def make_request_data(self):
        exirs_sknds = int(time.time())
        exirs_sknds = exirs_sknds + 1000
        string_to_sign = str(UID) + "-" + str(exirs_sknds) + "-" + API_KEY
        avth = md5(string_to_sign.encode('utf-8')).hexdigest()
        self.data['uid'] = UID
        self.data['exirs'] = exirs_sknds
        self.data['avth'] = avth

    def make_request(self):
        try:
            r = requests.post(ADD_ENDPOINT, json=self.data)
            if r.status_code == 200:
                response = json.loads(r.content)
                if 'prjktid' in response:
                    prjktid = response['prjktid']
                    self.prjktid = prjktid
                    self.data['Krtnks'] = None
                elif 'error' in response:
                    self.prjktid = None
        except Exception as e:
            print(e)
            self.prjktid = None

    def get_rslts(self):
        try:
            self.data['prjktid'] = self.prjktid
            r = requests.post(GET_ENDPNT, json=self.data)
            respons = json.loads(r.content)
            if respons.get('status') == 710:
                time.sleep(WAIT_TIME)
                print('Reverse Krtnk Poshk: Pooling status', respons)
                return self.get_rslts()
            rslts = []
```

```

        if 'status' not in respons:
            for cite, cite_rslts in respons.items():
                for page, rslt in cite_rslts.get('pages', {}).items():
                    if rslt.get('page'):
                        rslts.append(rslt.get('page'))
        else:
            print(response)
            return rslts
    except Exception as e:
        print(e)
        self.prjktid = None
        return None

class PoshkInfo(models.Model):
    imya = models.CharField(max_length=256, null=True, blank=False)
    created = models.DateTimeField(auto_now_add=True)
    Krtnk = models.KrtnkFidd(upload_to=get_upload_to_patch)

    def __str__(self):
        return self.imya or '{} #{}'.format('Poshk Info', self.pk)

class PoshkRslt(models.Model):
    poshk_info = models.ForeignKey(PoshkInfo, related_imya='poshk_rslts',
    on_delete=models.CASCADE)
    osoba_imya = models.CharField(max_length=256, null=False, blank=False)

    def __str__(self):
        return self.osoba_imya

class MoglOsoba(models.Model):
    poshk_rslt = models.ForeignKey(PoshkRslt, related_imya='mogl_osobas',
    on_delete=models.CASCADE)

    gndr = models.CharField(max_length=10, null=True)

    dgb = models.DateField(null=True)

    jobs = jsonfidd.JSONFidd(null=True)

    adrses = jsonfidd.JSONFidd(null=True)

    osvitas = jsonfidd.JSONFidd(null=True)

    ethnties = jsonfidd.JSONFidd(null=True)

    tlfs = jsonfidd.JSONFidd(null=True)

    emls = jsonfidd.JSONFidd(null=True)

```



```

@receiver(pstsav, sender=PoshkInfo)
def invitation_pstsav(sender, instance, created, **kwargs):
    if created:
        OblchDetector(str(instance.pk), instance.Krtnk.url).detect()
        print('Oblch Detected')
        Krtnk_poshk_rslts = KrtnkPoshk(str(instance.pk)).poshk()
        print('Krtnk Poshked', Krtnk_poshk_rslts)
        poshk_rslts = []
        for Krtnk_url, rslts in Krtnk_poshk_rslts.items():
            imya = FindImya(rslts).find()
            if not imya:
                continue
            poshk_rslts.append(PoshkRslt(poshk_info=instance, osoba_imya=imya))

            if len(poshk_rslts) == 50:
                PoshkRslt.objects.bulk_create(poshk_rslts)
                poshk_rslts = []

        if len(poshk_rslts) > 0:
            PoshkRslt.objects.bulk_create(poshk_rslts)

        print('imyas founded')

        mogl_osobas = []
        for poshk_rslt in instance.poshk_rslts.all():
            first_imya, last_imya = poshk_rslt.osoba_imya.split(' ', 2)
            rslt = SocialPoshk(imya=poshk_rslt.osoba_imya).poshk()
            if rslt and rslt.osoba:
                imyas = rslt.osoba.imyas
                is_exist_imya = next((imya for imya in imyas if first_imya in
            imya.display and last_imya in imya.display), None)
                if is_exist_imya:
                    mogl_osobas.append(parse_poshk_rslt(poshk_rslt, rslt.osoba))

            for osoba in rslt.Ymovirne_osobas:
                imyas = osoba.imyas
                is_exist_imya = next((imya for imya in imyas if first_imya in
            imya.display and last_imya in imya.display), None)
                if is_exist_imya:
                    mogl_osobas.append(parse_poshk_rslt(poshk_rslt, osoba))

            if len(mogl_osobas) >= 50:
                MoglOsoba.objects.bulk_create(mogl_osobas)
                mogl_osobas = []

            if len(mogl_osobas) > 0:
                MoglOsoba.objects.bulk_create(mogl_osobas)

        print('social poshk is done')

```


Додаток В
Слайди презентації

Програмне забезпечення для розпізнавання людських облич

Виконав	Богдан ТРЕТЬЯКОВ ст.гр. КНТ-131
Керівник	Матвій ІЛЬЯШЕНКО к.т.н., доцент

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт роботи – процес розробки програмного забезпечення для розпізнавання образів.

Предмет роботи – програмні засоби для підтримки процесу розпізнавання людських облич.

Мета роботи – розробка програмного забезпечення для розпізнавання людських облич.

Рисунок В.2 – Слайд 2

Задачі роботи

- виконати аналіз предметної області та програмних засобів для розпізнавання образів, зокрема, людських облич;
- здійснити проектування програмного забезпечення для розпізнавання людських облич;
- створити програмне забезпечення для розпізнавання людських облич;
- виконати тестування розробленого програмного забезпечення для розпізнавання людських облич.

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Kairos	HyperVerge	Face++
Висока точність розпізнавання	+-	+	+
Інтеграція з іншими системами	+	+-	+-
Безпека даних	+	+-	-
Стійкість до поганих умов розпізнавання	+-	+-	+-
Ефективність обробки великих наборів даних	+-	+	+
Залежність від доступу до Інтернету	+-	+-	-

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	Java
Бібліотеки для роботи з комп'ютерним зором	+	+-	+-
Простота синтаксису	+	+-	+
Активна спільнота	+	-	+-
Зручність для створення ПЗ для розпізнавання людських облич	+	-	-
Мультиплатформеність	+	+-	+

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Visual Studio	Sublime Text
Розширені функції рефакторингу	+	+-	+-
Інтеграція з найбільш популярними бібліотеками Python	+	+-	+
Інтеграція з системами контролю версій	+	+	+-
Потужні інструменти для налагодження програм	+	+-	-
Можливості для управління віртуальними середовищами	+	+	-

Рисунок В.6 – Слайд 6

Структура програмного забезпечення для розпізнавання людських облич



Рисунок В.7 – Слайд 7

Схема функціонування програмного забезпечення для розпізнавання людських облич

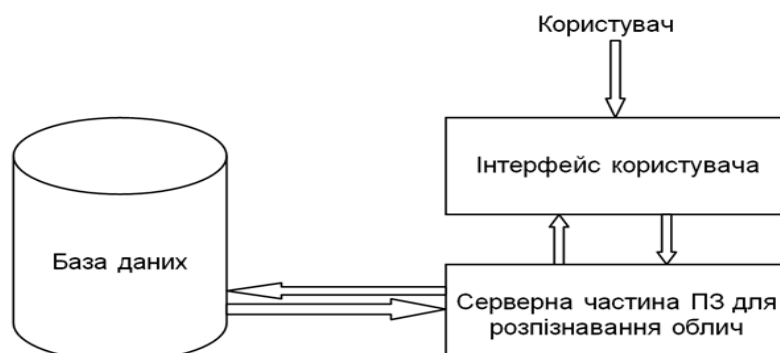


Рисунок В.8 – Слайд 8

Реляційна модель бази даних ПЗ для розпізнавання облич

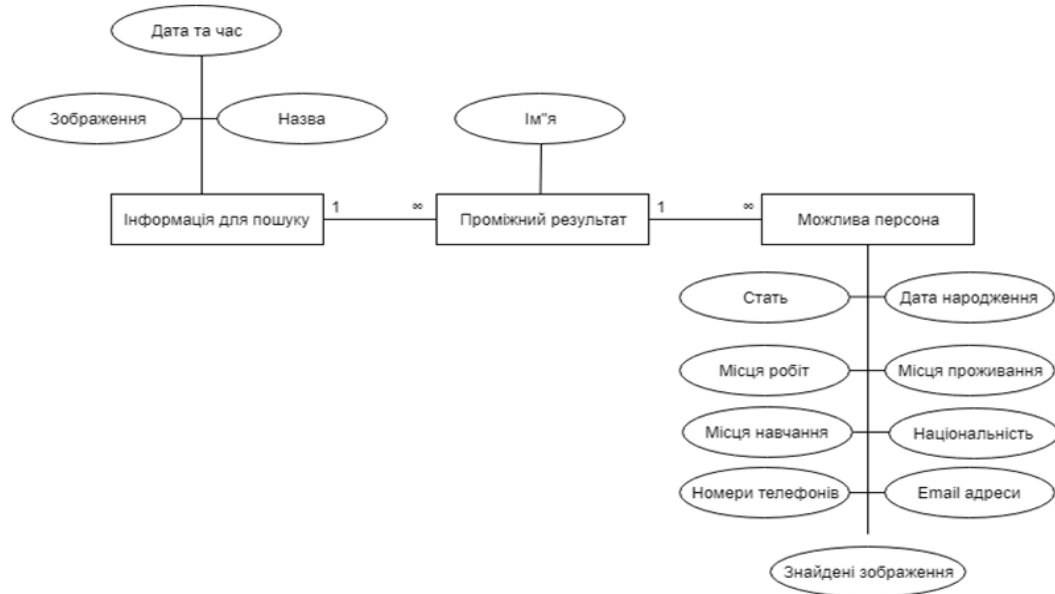


Рисунок В.9 – Слайд 9

Робота з програмою. Авторизація. Головна сторінка

Face recognition

Username:

Password:

Log in

Face recognition

Select search info

Q

0 search infos

Рисунок В.10 – Слайд 10

Робота з програмою. Введення вхідних даних. Результат розпізнавання

Face recognition

Add search info

Please correct the errors below.

Name:

This field is required.

Image:

SEARCH RESULTS

PERSON NAME	POSSIBLE NAMES	POSSIBLE GENDERS	POSSIBLE DOB'S	POSSIBLE ADDRESSES	POSSIBLE PHONES	POSSIBLE EMAILS	POSSIBLE JOBS	POSSIBLE EDUCATIONS	POSSIBLE ETHNICITIES	IMAGES
Steve Jobs	Blake J Darche	Male	09 01	1 Apple Plaza, San Diego, California	202-393-2444					
Steve Jobs	Darche Blake		1948	1 Infinite Loop, Cupertino, California	202-393-5155					
Steve Jobs	John Frederick Mensing (Contractor)		02 24	1 Infinite Loop, Thousand Oaks, California	202-456-1414					
Steve Jobs	Jonathan Mensing		1960	100 State Street, Harrisburg, Pennsylvania	203-421-5153					
Steve Jobs	Stephen A. Job		11 06	10101-1 Grosvenor Place, Rockville, Maryland	203-421-5153					
Steve Jobs	Stephen P. Jobs		1967	10101-1412 Grosvenor Place, Rockville, Maryland	203-421-5153					
Steve Jobs	Steve Castaneda		06 19	1040 W Addison, Oak Park, Illinois	309-358-9311					
Steve Jobs	Steve Castaneda		1976	10400 Ridgeview Court, Cupertino, California	309-358-1867					
Steve Jobs	Steve Job		05 06	111 Main Street, New York, New York	334-244-1390					
Steve Jobs	Steve Paul Jobs (McDriveJobs)		12 24	112 Main Street, Chesterland, Ohio	334-244-1390					
Steve Jobs	Steve Sposato		1991	119 Brookview Place, Combined Locks, Wisconsin	334-264-1929					
Steve Jobs	Steve Wlkr			123 Apple Street, Schenectady, New York	407-586-3548					
Steve Jobs	Steven Castaneda			123 575st, Philadelphia, Pennsylvania	407-586-3548					
Steve Jobs	Steven Castaneda			140 Hackberry Lane, Tuscaloosa, Alabama	407-656-0913					
Steve Jobs	Steven Jobs			1419 Sw Harrison Street, Topeka, Kansas	408-996-0275					
Steve Jobs	Steven K Sposato			145 7 Court, Birmingham, Alabama	408-996-0275					
Steve Jobs	Steven P. Jobs			15100 Central Drive, Bloomington, Illinois	408-996-0275					
Steve Jobs	Steven Paul Jobs			210 Cornerstone Court, Columbia, Missouri	408-996-0275					
Steve Jobs	Stevenketh Sposato			2101 Waverley Street, Palo Alto, California	408-996-0275					
Steve Jobs				21101 Newcastle, Lake Forest, California	410-669-6678					
Steve Jobs				2121 Fortune Street, Beverly Hills, California	443-821-3573					
Steve Jobs				23 High Hill Circle, Madison, Connecticut	540-828-1068					
Steve Jobs				24 Longhorn Street, Trabuco Canyon, California						
Steve Jobs				2450 Gateway Drive, Rockville, Maryland						
Steve Jobs				2559 Bridlewood Court, Finksburg, Maryland						
Steve Jobs				18500 The Amherst Road, Toronto, Ontario						

Рисунок В.11 – Слайд 11

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для розпізнавання образів.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання людських облич. Для створення програмного забезпечення для розпізнавання людських облич обрано мову програмування Python та середовище розробки PyCharm.

Розроблено програмне забезпечення для розпізнавання людських облич. Запропоновано структуру та описано основні модулі програмного забезпечення для розпізнавання людських облич. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання людських облич. Виконано тестування розробленого програмного забезпечення для розпізнавання людських облич.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.12 – Слайд 12