

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)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

ЗАСОБІВ ЛОКАЛІЗАЦІЇ ОБЛИЧ НА ЗОБРАЖЕННЯХ

RESEARCH AND SOFTWARE IMPLEMENTATION OF

FACE LOCALIZATION TOOLS IN IMAGES

Виконав: студент 2 курсу, групи КНТ-112м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КРИВА С.С.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
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
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КРИВИ Семена Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів локалізації обличчя на зображеннях. Research and Software Implementation of Face Localization Tools in Images

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 14 ” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 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	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 16 ” жовтня 2023 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	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Семен КРИВА
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
89 с., 36 рис., 3 табл., 3 дод., 17 джерел.

PYTHON, PYCHARM, ЛОКАЛІЗАЦІЯ ОБЛИЧ, МОВА
ПРОГРАМУВАННЯ, ОБРОБКА ЗОБРАЖЕНЬ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для обробки зображень.

Предмет дослідження – програмні засоби для локалізації облич на зображеннях.

Метою роботи є дослідження та розробка програмного забезпечення для локалізації облич на зображеннях.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm, система керування базами даних SQLite.

Результати. Розроблено програмне забезпечення для локалізації облич на зображеннях за допомогою мови програмування Python та середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного забезпечення, що реалізує методи локалізації облич на зображеннях та дозволяє виявляти контури облич на заданих зображеннях.

Висновки. Виконано проєктування програмного забезпечення для локалізації облич на зображеннях. Розроблено програмне забезпечення для локалізації облич на зображеннях. Здійснено тестування розробленого програмного забезпечення для локалізації облич на зображеннях.

Галузь використання – програмні засоби верифікації осіб.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 36 figures, 3 tables, 3 appendixes, 17 sources.

PYTHON, PYCHARM, FACE DETECTION, PROGRAMMING LANGUAGE, IMAGE PROCESSING, SOFTWARE.

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

The subject of the research is software for localization of faces in images.

The purpose of this work is to develop the software for localization of faces in images.

Materials, methods and technical tools: Python programming language, PyCharm development environment, SQLite database management system.

Results. The software tools for face localization in images using Python programming language and PyCharm development environment have been developed.

The practical value of the work lies in the development of software that implements methods of face localization in images and allows you to detect of contours of faces on given images.

Conclusions. The software for localization of faces in images has been designed. The software for localization of faces in images has been developed. The developed software for localization of faces in images has been tested.

Scope of use – personal verification software.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про локалізацію облич на зображеннях	11
1.2 Аналіз програмного забезпечення для локалізації облич на зображеннях	23
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування.....	34
2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для локалізації облич на зображеннях	37
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях.....	39
2.4 Висновки за розділом 2	41
3 Опис програми	43
3.1 Структура програмного забезпечення для локалізації облич на зображеннях	43
3.2 Функціонування програмного забезпечення для локалізації облич на зображеннях	47
3.3 Проєктування програмного забезпечення для локалізації облич на зображеннях	48
3.4 Проєктування інтерфейсу програмного забезпечення для локалізації облич на зображеннях.....	50
3.5 Висновки за розділом 3	53
4 Експлуатація, тестування та експериментальне дослідження програми.....	54
4.1 Призначення й умови застосування програми	54
4.2 Характеристики програми для локалізації облич на зображеннях	55
4.3 Інструкція по експлуатації програми.....	55

4.3.1	Звернення до програми.....	55
4.3.2	Вхідні й вихідні дані.....	55
4.3.3	Повідомлення.....	56
4.4	Виконання програмного забезпечення для локалізації облич на зображеннях	56
4.5	Тестування програмного забезпечення для локалізації облич на зображеннях	59
4.6	Висновки за розділом 4	61
	Висновки.....	62
	Перелік джерел посилання	65
	Додаток А Технічне завдання.....	67
	Додаток Б Фрагмент тексту програми	71
	Додаток В Слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CNN – Convolutional Neural Network, згорткова нейронна мережа;

HOG – Histogram of Oriented Gradients, гістограма спрямованих градієнтів;

IDE – Integrated Development Environment, інтегроване середовище для розробки;

PyCharm – інтегроване середовище розробки для мови програмування Python;

UML – Unified Modeling Language, уніфікована мова моделювання;

ООП – об'єктно-орієнтоване програмування;

ПЗ – програмне забезпечення.

ВСТУП

Використання програмних засобів для локалізації облич на зображеннях є дуже актуальним і має великий потенціал у різних сферах. Однією з головних причин актуальності є зростання використання комп'ютерного зору, штучного інтелекту та машинного навчання в різних галузях [1]-[4]. Так, у сфері безпеки та відео спостережень системи локалізації облич допомагають відслідковувати та розпізнавати осіб на відеозаписах, що сприяє виявленню злочинців та запобіганню злочинам. У медичній галузі такі програми можуть бути використані для автоматичної ідентифікації пацієнтів та моніторингу їхнього стану на основі медичних зображень [1]-[4]. Крім того, ця технологія має великий потенціал у сфері розваг та реклами, оскільки програми ідентифікації облич можуть впізнавати обличчя користувачів і пропонувати персоналізований контент або рекламу [1]-[4].

Крім того, зростаюча доступність потужних обчислювальних ресурсів та постійний розвиток технологій комп'ютерного зору роблять використання програмних засобів для локалізації облич на зображеннях дуже актуальним для різних галузей, включаючи безпеку, медицину, рекламу та розваги [1]-[4].

Проте впровадження та підтримка деяких програм для локалізації облич на зображеннях може вимагати значних фінансових і технічних витрат на навчання моделей, розробку та підтримку інфраструктури. У деяких випадках програм для локалізації облич на зображеннях можуть використовувати штучні обхідні методи для покращення точності, що може спотворити результати і робити їх непередбачуваними. Деякі програми можуть виявляти більш високу точність при розпізнаванні облич певних етнічних або культурних груп порівняно з іншими, що може призвести до некоректного розпізнавання або дискримінації [1]-[4]. Тому актуальною є розробка програмного забезпечення для локалізації облич на зображеннях, що використовує різні методи виявлення та розпізнавання облич.

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки програмного забезпечення для локалізації облич на зображеннях.

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

- виконати аналіз предметної області та програмних засобів для обробки зображень шляхом їх сегментації;
- розробити модель взаємодії користувача з програмним забезпеченням для сегментації зображень;
- здійснити проєктування програмного забезпечення для сегментації зображень;
- створити програмне забезпечення для сегментації зображень;
- виконати тестування розробленого програмного забезпечення для сегментації зображень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про локалізацію облич на зображеннях

У наш час розпізнавання облич оцінюється як одне з найперспективніших завдань аналізу зображень. Локалізація (детекція) облич є важливою частиною завдання розпізнавання облич. Локалізації облич на фотографіях перешкоджають відмінності людських облич у позах, виразах, положенні та орієнтації, кольорі шкіри, наявності окулярів або волосся на обличчі, посиленні камери, умовах освітлення та роздільній здатності зображення [1].

Локалізація об'єктів являє собою комп'ютерну технологію в поєднанні з обробкою зображень і комп'ютерним зором для виявлення людських облич, будівель, дерев, автомобілів тощо. Основною метою алгоритму локалізації обличчя є його ідентифікація на зображенні [1]-[7].

Сьогодні активно проводиться велика кількість досліджень у галузі впізнавання осіб та локалізації облич, з метою підвищити ефективність процесу локалізації та розпізнавання [1].

Процес локалізації облич є першим та ключовим етапом в широкому контексті розпізнавання осіб, і його застосування розповсюджується на різноманітні зображення. У сферах забезпечення безпеки, біометрії, правоохоронних органів, розваг, особистої безпеки і багатьох інших локалізація облич виявляється дуже корисною та необхідною [1].

Ця технологія дозволяє здійснювати розпізнавання облич в реальному часі, відстежуючи рухи та місцезнаходження людей чи предметів. Вона широко використовується в мобільних камерах та фотокамерах для створення різних кадрів. Наприклад, Facebook використовує алгоритми розпізнавання облич для ідентифікації осіб на фотографіях [1].

Проаналізуємо підходи до локалізації облич на зображеннях. Групи методів ідентифікації облич можна розділити на чотири основні категорії, і алгоритми локалізації облич можуть належати до однієї або навіть кількох з

цих груп [1]-[7]. Класифікація методів локалізації облич може бути подана таким чином [1]:

– методи на основі знань. Методи цієї категорії використовують правила та знання, побудовані на основі інтуїції та досвіду людей, що використовуються для локалізації та розпізнавання облич. Вони аналізують передні обличчя, розділяючи ніс, очі та рот у декількох позиціях. Однак головним недоліком таких методів є важкість управління правилами: якщо правила занадто загальні або докладні, може виникнути багато помилкових відповідей. Цей підхід недостатньо ефективний і не здатний ефективно розпізнавати обличчя на багатьох зображеннях [1];

– методи на основі ознак. Ця категорія методів базується на виявленні особливостей обличчя шляхом виділення структурних компонентів. Спочатку модель навчається як класифікатор, а потім використовується для виділення областей обличчя та інших областей. Мета полягає в розпізнаванні обличчя з урахуванням навіть незвичних особливостей. Цей метод розбивається на етапи, де кілька зразків обличчя досягають досить високої успішності [1];

– методи шаблонного співставлення. Ці методи використовують попередньо визначені або параметризовані шаблони обличчя для пошуку або виявлення облич на основі взаємозв'язку між шаблонами та вхідними зображеннями. Наприклад, обличчя можна розділити на окремі риси, такі як очі, ніс та рот. Можна також використовувати методи виявлення країв для створення моделі обличчя. Цей підхід має просту реалізацію, але не завжди ефективно розпізнає обличчя, і для його вдосконалення використовують деформаційні методи [1];

– методи на основі зовнішності. Методи цієї категорії спираються на навчені представлення зображень обличчя для розпізнавання його рис. Цей підхід, заснований на вигляді, є більш продуктивним в порівнянні з іншими. Зазвичай методи зовнішності використовують статистичний аналіз та машинне навчання для виявлення важливих рис обличчя на зображеннях. Цей метод є ефективним при розпізнаванні рис обличчя [1].

Методи локалізації облич на основі зовнішності можна розподілити таким чином [1]:

- методи на основі алгоритму аналізу головних компонентів;
- методи на основі розподілу. Алгоритми, такі як дискримінант Фішера, можуть бути використані для визначення підпросторів, що представляють шаблони обличчя. Внаслідок цього синтезується навчений класифікатор, який точно ідентифікує екземпляри класу цільової моделі [1];
- методи на основі використання нейронних мереж. За допомогою нейронних мереж можна ефективно вирішувати різноманітні завдання розпізнавання, такі як локалізація об'єктів, виявлення облич, емоцій та розпізнавання інших образів [1];
- використання методу опорних векторів. Мащини опорних векторів є лінійними класифікаторами, що максимізують розрив між гіперплощиною рішення та прикладами в навчальному наборі [1];
- застосування розрідженої мережі. Цей метод включає створення розрідженої мережі, що складається з двох лінійних компонентів або вузлів: один представляє модель з обличчям, а інший – модель без нього. Цей підхід відзначається більшою ефективністю та скороченим часом обробки [1];
- прості наївні байєсівські методи. Для визначення ймовірності наявності обличчя на зображенні застосовується підрахунок кількості входжень різних шаблонів на навчальних зображеннях. Класифікатор зберігає статистику про розташування та характеристики локальних облич [1];
- використання прихованих моделей Маркова. Цей метод визначає простір моделі, враховуючи особливості обличчя, які часто виражаються піксельними лініями. Приховані моделі Маркова також активно використовуються для розробки інших пошукових алгоритмів [1];
- теоретичні підходи до обробки даних. Використання випадкових полів Маркова може бути варіантом для аналізу візерунків обличчя та їхніх асоційованих рис. Процес Маркова здійснює максимізацію розрізненості між

класами використовуючи спеціальні критерії, що дозволяють використовувати цей підхід для локалізації облич [1];

– підхід на основі індуктивного навчання. Цей підхід використовується для розпізнавання облич та використовує алгоритми, такі як C4.5 Quinlan або інші методи синтезу моделей на основі дерев розв'язків [1]-[7].

Локалізація обличчя базується на різноманітних методах, які допомагають забезпечити точне впізнавання осіб. Ці підходи включають в себе використання різних програмних бібліотек, нейронних мереж та інших методів обробки зображень. Ці методи часто мають схожі етапи для проведення процесу локалізації обличчя [1]-[7].

Механізм локалізації обличчя відповідає за визначення осіб на зображеннях так само, як і визначення кількох об'єктів на одному зображенні [1]-[7].

Проаналізуємо етапи функціонування програмного забезпечення для локалізації облич на зображеннях [1]-[7].

Робота програмного забезпечення для локалізації облич на зображеннях починається з імпорту зображення шляхом вказівки місцезнаходження файлу [1].

На наступному етапі зображення перетворюється з формату RGB до градацій сірого, це допомагає зосередитися на деталях обличчя [1].

Далі зображення піддається обробці, включаючи зміну розміру, обрізку, розмиття та підсилення контрастності, якщо необхідно [1].

Після цього застосовується сегментація зображення, яка допомагає виділити контури обличчя або інших об'єктів на зображенні, забезпечуючи класифікатору можливість швидко ідентифікувати особи та об'єкти на фото [1].

Наступним етапом є використання алгоритму, який нагадує методику розпізнавання обличчя за аналогією з алгоритмом Віола-Джонса [1]. Це дозволяє визначити положення обличчя людини на фотографії чи зображенні. Всі обличчя мають спільні риси, які характерні для людських облич, такі як

темніша область очей порівняно з оточуючими пікселями та яскравіша область носа порівняно з областю очей [1].

Подібний до методу Хаара алгоритм також може бути використаний для виділення об'єкту або функцій на зображенні за допомогою визначення різниць у контурі зображення, виявлення ліній, очей, носа, рота тощо, ці контури використовуються для ідентифікації облич. Це дозволяє виділяти ключові характеристики зображення та їх подальше використання для розпізнавання обличчя [1].

Далі важливим кроком є визначення координат, які використовуються для створення прямокутника на зображенні, що демонструє положення обличчя. Ця область також може називатися зоною інтересу на зображенні. За допомогою цих координат можна знайти контур відповідної форми та побудувати прямокутник. Крім того, існують інші методи виявлення, які можуть бути поєднані для визначення таких рис, як посмішка, очі, моргання тощо. Розглядається також можливість використання інших методів [1].

У наш час для локалізації облич на зображеннях широко використовується машинне навчання [2], що є галуззю штучного інтелекту, яка займається проектуванням і розробкою алгоритмів. Машинне навчання дозволяє комп'ютерам навчатися з даних, розробляючи алгоритми, які автоматично виділяють шаблони у даних та роблять передбачення на їх основі [2]. Алгоритми машинного навчання навчаються на даних, щоб вирішувати завдання, які є надто складними для традиційного програмування [2].

Глибоке навчання входить до складу машинного навчання і охоплює декілька алгоритмів, які працюють спільно. Важливо зазначити, що терміни "машинне навчання" та "глибоке навчання" часто використовуються як синоніми. На сьогоднішній день більшість розвитку у галузі машинного навчання відбувається на рівні глибокого навчання [2].

Локалізація обличчя є біометричною технологією, яка використовує унікальні риси обличчя для ідентифікації особи. Більшість систем

розпізнавання обличчя операційно працюють за допомогою порівняння особливостей обличчя зі збереженими в базі даних. У випадку відповідності, система розпізнає особу. Однак, якщо обличчя не знаходиться в базі даних, система не може впізнати особу [2].

Технологія локалізації обличчя зазвичай використовується з метою забезпечення безпеки, наприклад, для виявлення злочинців або захисту особистих даних від крадіжок. Її також можна використовувати для різних повсякденних завдань, наприклад, пошуку загубленої дитини в людних місцях або ідентифікації видатних осіб на подіях [2].

Деякі системи локалізації обличчя використовують штучний інтелект, який може навчитися впізнавати зміни у зовнішності людини, такі як нарощування бороди або приріст ваги [2].

Застосування машинного навчання в технології локалізації обличчя. Один з найпоширеніших методів машинного навчання для локалізації обличчя – це згортова нейронна мережа глибокого навчання (convolutional neural network, CNN). CNN є типом штучної нейронної мережі, яка особливо підходить для завдань локалізації зображень [2].

Згорткові нейронні мережі здатні навчатися виділяти характеристики зі зображень та класифікувати їх за певними категоріями, використовуючи ці характеристики. Глибина такої мережі має ключове значення для визначення обличчя, оскільки глибокі шари в CNN спроможні розпізнавати більш складні аспекти обличчя [2].

Для прикладу, поверхневий шар CNN може навчитися визначати базові ознаки обличчя, такі як форма носа або розташування очей. У той час як глибокий шар CNN може вчитися розпізнавати більш вдосконалені аспекти, такі як відтінки шкіри чи контури щелепи. Коли нейромережа CNN оптимально навчена використовувати вхідні зображення обличчя, її можна використовувати для локалізації обличчя на нових зображеннях [2].

Процес розпізнавання обличчя може бути розділений на три послідовні етапи [2].

Першим важливим етапом є виявлення обличчя на початковому зображенні. Цю задачу можна виконати за допомогою каскадних класифікаторів, які навчені на позитивних та негативних прикладах. Такі класифікатори використовуються в різних системах, включаючи вбудовані функції розпізнавання обличчя у камерах, а також платформи соціальних мереж, наприклад Snapchat та Facebook, для додавання ефектів до фото і відео [2].

Однією з проблем локалізації обличчя є те, що обличчя може бути спрямоване в інший бік від камери. Ця різниця в орієнтації впливає на розпізнавання. Одним із способів рішення цієї проблеми є нормалізація обличчя, перетворюючи його на стандартний вигляд. Для цього можна використовувати загальні ознаки, такі як положення верхньої губи, носа, очей, рота тощо. Це дозволяє нормалізувати різні обличчя до спільної базової форми [2].

Наступним етапом є виділення розмірів та ознак об'єктів. Після виявлення і вирівнювання обличчя використовується згортоква нейронна мережа (CNN) для виділення характеристик високого рівня з зображення. Ці характеристики далі використовуються для подальшого розпізнавання облич з бази даних [2].

Останнім етапом є розпізнавання обличчя. На цьому етапі виділені характеристики об'єднуються з відповідним обличчями у базі даних. Зазвичай для порівняння використовується метрика евклідової відстані, яка оцінює схожість між двома векторами, що представляють риси обличчя [2].

Розпізнавання обличчя є методом біометрії, який використовується для ідентифікації осіб за їхніми рисами обличчя на основі зображень або відео та порівняння цих даних з наявною базою відомих облич, з метою знаходження відповідностей [2].

Технологія локалізації обличчя використовує біометричні дані та розуміння облич людей, зазвичай за допомогою штучного інтелекту. Цей процес полягає у виділенні рис обличчя з фотографій або відео та порівнянні

їх зі збереженими даними відомих облич, щоб знайти відповідності. Розпізнавання обличчя може допомогти підтвердити особу, однак такий підхід також породжує питання про приватність [2].

Процес локалізації та розпізнавання облич може бути виконаний за допомогою спеціального програмного забезпечення таким чином. Спеціальне програмне забезпечення містить зображення або відео з обличчям людини [2]. Програма аналізує ці візуальні дані, створює розклад обличчя людини, який називається підпис обличчя. Цей підпис включає різноманітні параметри, такі як положення очей, можливі родимки або особливості обличчя [2]. Для порівняння підпису обличчя із відомими даними використовується система розпізнавання обличчя [2]. Система розпізнавання обличчя порівнює підпис обличчя з даними в базі та визначає, чи існують відповідності. Деякі системи також можуть розраховувати показники точності та надавати можливі варіанти [2].

Системи локалізації обличчя реєструють вирази обличчя: вони базуються на алгоритмах, а не на мозку, і використовуються для ідентифікації осіб. Там, де людина бачить обличчя, технологія розпізнавання реєструє дані. Ці дані можуть бути збережені та використовувані [2].

Згідно з різними дослідженням доволі значна частка дорослих громадян розвинених країн додала свої фотографії до однієї або кількох баз даних для розпізнавання облич, які використовуються правоохоронними органами [2]. Використання систем розпізнавання облич у мобільних телефонах та практика сканування соціальних мереж для навчання алгоритмам, безперечно, збільшили цей показник [2].

Системи локалізації обличчя застосовуються повсюди, від мобільних телефонів до систем відео спостереження в магазинах. Проаналізуємо різні сфери діяльності, де розпізнавання та локалізація облич допомагає спростити життя та забезпечити безпеку [2].

– аеропорти: технологія розпізнавання та локалізації облич використовується для виявлення осіб, які порушили умови візи або можуть

бути пов'язані зі злочинами. Деякі авіакомпанії також використовують розпізнавання обличчя для швидкого реєстраційного процесу [2];

- мобільні телефони: розпізнавання обличчя використовується для розблокування деяких телефонів, а також для відкриття цифрового гаманця. Імовірність помилкового розблокування обличчям в Apple становить один до мільйона [2];

- освіта: програмні системи розпізнавання та локалізації обличчя впроваджуються в закладах освіти для підвищення безпеки та контролю присутності [2];

- соціальні мережі: Facebook використовує алгоритми розпізнавання обличчя для автоматичного позначення людей на завантажених фотографіях. Ця технологія має точність близько 98% [2];

- корпоративні середовища: деякі компанії використовують розпізнавання обличчя для заміни пластикових карток доступу. В магазинах роздрібної торгівлі системи розпізнавання використовуються для виявлення потенційних злодіїв [2];

- маркетинг: маркетологи можуть використовувати дані з розпізнавання обличчя для налаштування рекламних кампаній, визначаючи цільові аудиторії за віком, статтю та етнічною приналежністю [2].

Перевагами використання програмних засобів для локалізації та розпізнавання обличчя є такі [2]:

- забезпечення безпеки польотів: аеропорти використовують розпізнавання обличчя, щоб ідентифікувати злочинців або потенційних загроз [2];

- можливість виявлення злочинців: програмні засоби локалізації та розпізнавання обличчя можна використовувати для ідентифікації підозрюваних за фотографіями чи відео [2];

- можливість захисту мобільного телефону. Багато сучасних телефонів використовують розпізнавання обличчя, щоб розблокувати або підтвердити особу перед покупкою [2].

Важливо розуміти обмеження та недоліки, які властиві програмним системам локалізації та розпізнавання облич [2]:

- помилкова ідентифікація: якщо використовувати лише програмні системи локалізації та розпізнавання облич при виявленні злочинців (без верифікації результату іншими засобами), то можуть виникнути ситуації, пов'язані з неправильною ідентифікацією осіб, відображаючи невинних людей як підозрілих [2];

- неточність результатів, пов'язана з віком людини. З віком людини точність програмних систем локалізації та розпізнавання облич може зменшуватися, особливо у випадку людей похилого віку [2];

- расові та гендерні упередження: дослідження свідчать, що програмні системи локалізації та розпізнавання облич можуть показувати недостатню ефективність у визначенні статі та розпізнаванні осіб різних расових груп [2];

- чутливість до змін: використання масок, сонцезахисних окулярів або макіяжу може спотворити точність результату програмної системи локалізації та розпізнавання облич [2];

- порушення конфіденційності: багато експертів висловлюють занепокоєння стосовно вторгнення у приватне життя через програмні системи локалізації та розпізнавання облич [2];

- загроза безпеці: дані про обличчя особи можуть бути збережені та використовуватися без її дозволу, що створює ризик для злому та викрадення зловмисниками її персональних, зокрема фінансових, даних [2];

- проблеми з правом власності: особиста інформація, зокрема зображення обличчя, може бути збережена і використовуватися в соціальних мережах або інших місцях, порушуючи права на конфіденційність та власність особи [1]-[7].

Програми для локалізації облич використовують штучний інтелект, навчання з вчителем, статистичний аналіз та алгоритми обробки зображень з метою знаходження людських облич на великих фотографіях і виділення їх серед інших ділянок, таких як пейзажі, будівлі та інші частини тіла. Перед

виконанням розпізнавання облич цифрові дані піддаються попередній обробці для покращення якості та видалення зображень, які можуть заважати процесу локалізації та розпізнавання [1]-[7].

Типовий підхід до алгоритмів локалізації облич полягає у пошуку людських очей, які є одними з найлегших для впізнавання елементів. Далі алгоритми стараються ідентифікувати інші характеристики обличчя, такі як брови, губи, ніздрі, область навколо очей тощо. Після того, як алгоритм зробив припущення про те, що він виявив обличчя, він виконує додаткову перевірку для впевненості у своїй знахідці [1]-[7].

Для досягнення високої точності алгоритм навчається на великих наборах даних, що містять велику кількість позитивних і негативних зображень. Це навчання сприяє покращенню здатності алгоритму визначати, чи присутні обличчя на зображенні та де саме вони знаходяться [1]-[7].

Програмне забезпечення для розпізнавання обличчя використовує різноманітні методи, кожен зі своїми перевагами та недоліками [1]-[7].

Метод Віолі-Джонса базується на навчанні моделі для розпізнавання, що є обличчям, а що ні. Хоча цей метод ефективний для локалізації та розпізнавання облич в реальному часі, він може мати певні труднощі з виявленням прихованих облич або обманливих зображень [1]-[7].

Підхід, заснований на знаннях або правилах включає в себе методи, які описують обличчя на основі заданих правил. Однак створення послідовних правил на основі знань може бути складним завданням [1]-[7]. Підхід, заснований на функціях, використовує методи, які розпізнають обличчя за конкретними ознаками, наприклад, очима чи носом. Однак шум і різна освітленість можуть негативно вплинути на їхню точність [1]-[7].

Шаблонний підхід ґрунтується на порівнянні зображення з попередньо збереженими стандартними моделями обличчя або їх характеристиками, а потім комбінує обидва підходи для локалізації та розпізнавання облич. Тим не менше, такий підхід може стикається з труднощами врахування відмінностей у положенні, масштабі та форм [1]-[7].

Підхід на основі зовнішнього вигляду використовує статистичний аналіз та машинне навчання для виявлення важливих особливостей на зображеннях обличчя. Методи, що базуються на зовнішньому вигляді, можуть ураховувати зміни в освітленні та напрямку [1]-[7].

Проблеми використання методів локалізації та розпізнавання обличчя спричиняють погіршену точність та відмінності в швидкості розпізнавання обличчя. Ці проблеми включають складний фон, наявність кількох обличчя на одному зображенні, незвичайні вирази обличчя, недостатнє освітлення, низьку роздільну здатність, перекриття обличчя, різний колір шкіри, відстань, орієнтацію та інше [1]-[7]. Проаналізуємо проблеми використання методів локалізації та розпізнавання обличчя [1]-[7]:

- незвичайні вирази обличчя: обличчя людини на фотографії може мати непередбачуваний або дивний вираз, що ускладнює використання деяких методів локалізації обличчя;

- освітлення: деякі частини зображення можуть бути перепалі або занадто темні через недостатнє або зайве освітлення.

- різні відтінки шкіри: розпізнавання обличчя різного кольору шкіри є викликом, що потребує багато навчальних зображень для досягнення прийнятної точності;

- віддаленість: якщо відстань велика, розмір об'єкта (у цьому випадку обличчя) може здається меншим і менш інформативним;

- положення та орієнтація: положення та кут обличчя відносно камери мають значний вплив на швидкість та якість його локалізації;

- складний фон: наявність інших об'єктів або осіб на зображенні ускладнює точність розпізнавання;

- багато обличчя на одному зображенні: розпізнавання обличчя, коли на зображенні присутні кілька осіб, вимагає додаткових зусиль для досягнення високої швидкості;

– загородження обличчя: Часткова закупорка обличчя окулярами, шарфами, руками, волоссям чи головними уборами може ускладнити процес розпізнавання;

– низька роздільна здатність: Зображення з низькою роздільною здатністю або великою кількістю шуму негативно впливають на точність розпізнавання [1]-[7].

Визначено, що локалізація облич є важливою частиною завдання розпізнавання облич та широко використовується у різних сферах людської діяльності, зокрема, в медицині, безпеки та відео спостережень, розваги та реклами. Проаналізовано підходи до локалізації облич на зображеннях. Виконано класифікацію методів локалізації облич. Досліджено етапи функціонування програмного забезпечення для локалізації облич на зображеннях. Описано використання методів та моделей машинного та глибокого навчання для локалізації облич. Проаналізовано сфери діяльності, де розпізнавання та локалізація облич допомагає спростити життя та забезпечити безпеку. Визначено переваги та недоліки використання програмних засобів для локалізації та розпізнавання облич. Проаналізовано проблеми використання методів локалізації та розпізнавання облич.

1.2 Аналіз програмного забезпечення для локалізації облич на зображеннях

Проаналізуємо програмне забезпечення для локалізації облич на зображеннях [8]-[13].

Програмне забезпечення OpenBR призначено для виконання локалізації облич на зображеннях [11]. Програмне забезпечення OpenBR (рис. 1.1) є безкоштовним програмним забезпеченням для виявлення облич, яке сприяє розробці відкритих алгоритмів та відтворюваних оцінок. Це програмне забезпечення підтримує різні операційні системи, зокрема, Windows, Linux, OS X та Raspbian [10], [11].

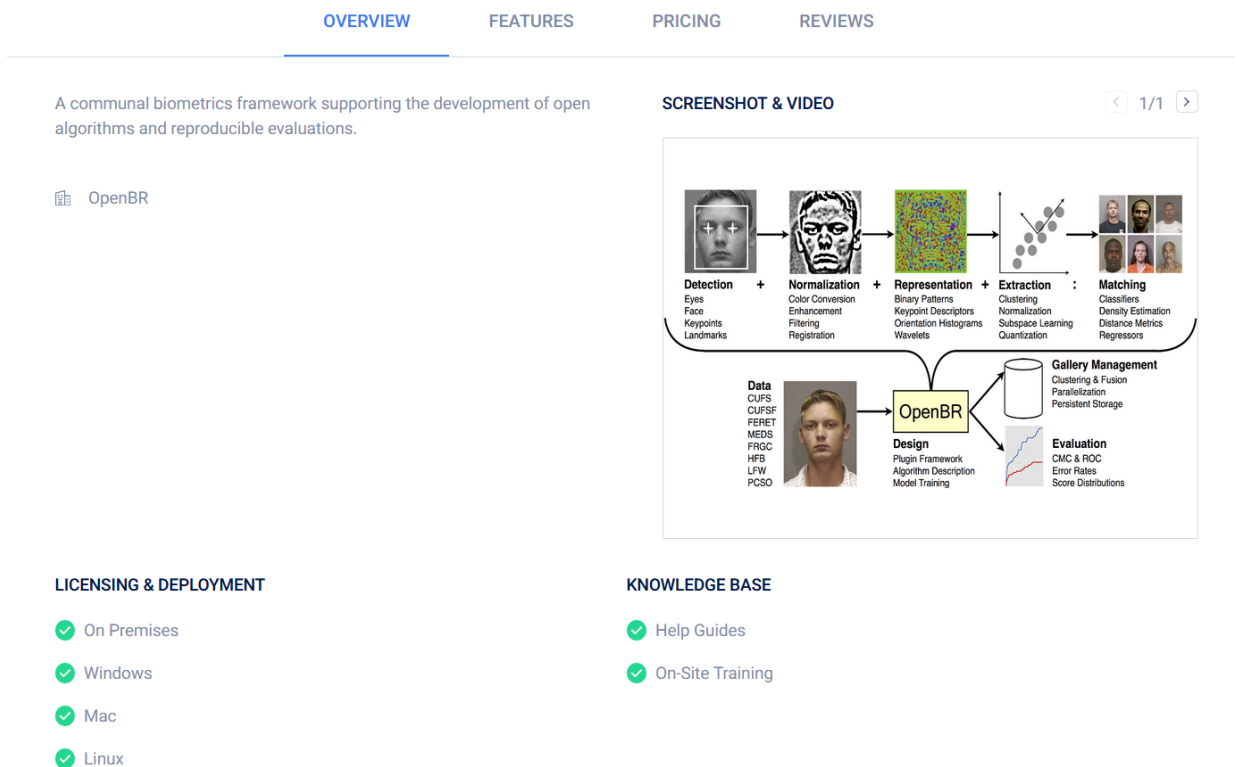


Рисунок 1.1 – Програмне забезпечення OpenBR [11]

Серед особливостей програмного забезпечення OpenBR для локалізації облич на зображеннях можна виділити такі [10], [11]:

– підтримка різних засобів інтеграції з іншим програмним забезпеченням. Програмне забезпечення OpenBR надає C++ API, яке може бути легко вбудоване у різні програми та проєкти, дозволяючи розробникам інтегрувати функціональність розпізнавання обличчя у свої програмні застосунки різного призначення [10], [11];

– OpenBR є повноцінним завершеним програмним комплексом, який охоплює не тільки локалізацію та розпізнавання облич, але й інші аспекти, такі як виявлення облич, позначення землі та інші. Це робить його універсальним інструментом для обробки зображень [10], [11];

– OpenBR є сумісним з вимогами та стандартами Національного інституту стандартів та технологій, що дозволяє використовувати його для оцінювання та порівняння різних алгоритмів локалізації та розпізнавання облич [10], [11];

– використання алгоритм 4SF2. Програмне забезпечення OpenBR використовує алгоритм 4SF2 для виконання розпізнавання облич. Цей алгоритм є ключовим компонентом програмної системи та допомагає досягти високої точності розпізнавання [10], [11];

– засоби оцінювання віку та статі людини. Поміж інших можливостей, програмні алгоритми OpenBR можуть використовуватися для оцінки віку та статі осіб на зображеннях [10], [11];

– підтримка різних операційних систем. Програмне забезпечення OpenBR підтримує різні операційні системи, такі як Windows, Linux, OS X та Raspbian. Це робить його доступним для розробників програмного забезпечення на різних платформах [10], [11];

– відкритий код. Програмне забезпечення OpenBR має відкритий вихідний код, що означає, що розробники програм можуть переглядати, модифікувати та вдосконалювати його функціональність відповідно до своїх потреб [10], [11];

– розширюваність: додаткові алгоритми розпізнавання обличчя та інші функції можуть бути додані до програмного забезпечення OpenBR, що дозволяє розробникам створювати власні рішення на основі цієї програмної платформи [10], [11];

– активна спільнота: програмне забезпечення OpenBR має активну спільноту розробників та користувачів, яка сприяє обміну досвідом, вирішенню проблем і вдосконаленню програмного забезпечення [10], [11];

– засоби автоматичного визначення особи: окрім локалізації облич, програмне забезпечення OpenBR також може автоматично визначати конкретну особу на зображенні, що робить його корисним для систем ідентифікації [10], [11];

– підтримка багатьох форматів зображень: програмне забезпечення OpenBR підтримує різні формати зображень, що дозволяє обробляти різноманітні файли без необхідності їх конвертації [10], [11].

Загалом, програмне забезпечення OpenBR є потужним інструментом для локалізації облич на зображеннях, який надає можливості виявлення та розпізнавання облич, оцінки віку та статі, і має відкриту архітектуру для інтеграції з іншим програмним забезпеченням [10], [11].

Серед недоліків програмного забезпечення OpenBR можна відзначити те, що це програмне забезпечення є досить його складним в налаштування, що вимагає певних технічних знань для ефективного використання. Крім того, для ефективної роботи програмного забезпечення OpenBR може бути необхідний потужний апаратний комплекс, особливо при обробці великого обсягу зображень [10], [11].

Програмне забезпечення OpenFaceTracker призначено для виконання локалізації облич на зображеннях, а також для відстеження та аналізу рухів обличчя на відеозаписах або в реальному часі [10], [12]. Програмне забезпечення OpenFaceTracker (рис. 1.2) використовується для автоматичного виявлення, відстеження та аналізу різних аспектів обличчя людини в руху, таких як вирази, позиція, рухи очей та інші деталі [10], [12].

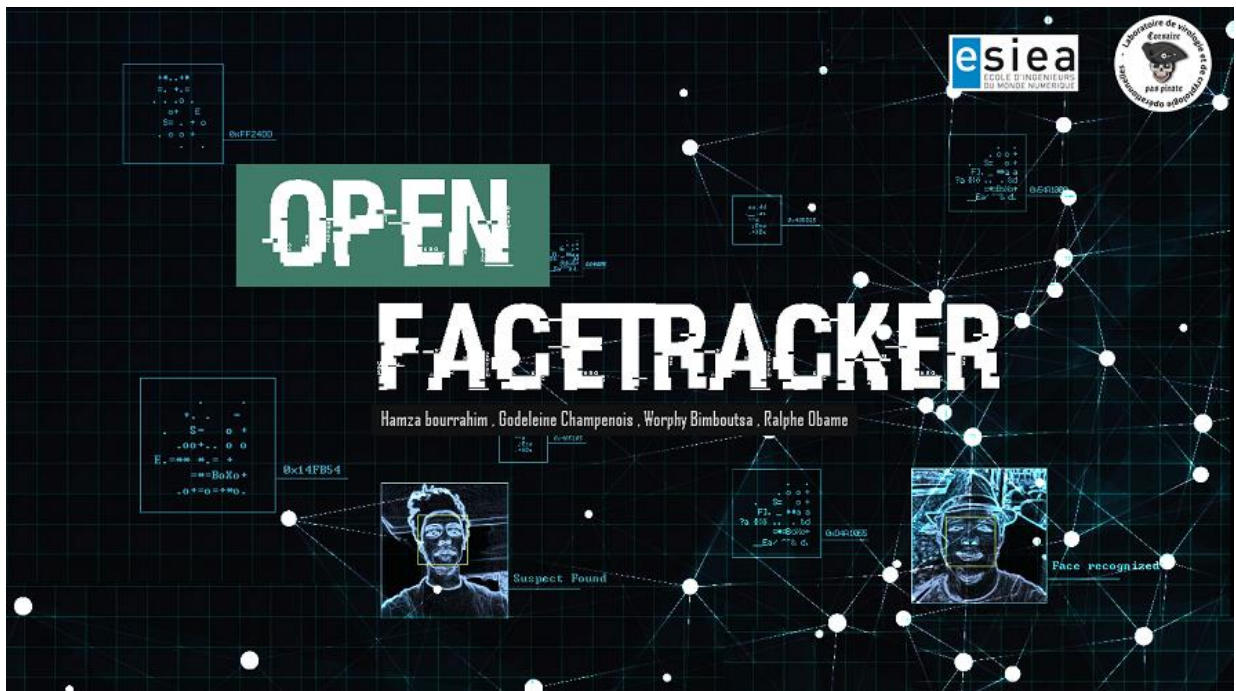


Рисунок 1.2 – Програмне забезпечення OpenFaceTracker [10], [12]

Серед особливостей програмного забезпечення OpenFaceTracker для локалізації облич на зображеннях можна виділити такі [10], [12]:

- засоби аналізу емоцій та виразів обличчя: програмне забезпечення OpenFaceTracker може виявляти вирази обличчя та емоції людини, що допомагає в розумінні настрою та реакцій людини на певні події або ситуації [10], [12];

- засоби дослідження рухів очей. Програмне забезпечення OpenFaceTracker може аналізувати рухи очей, включаючи фіксацію, рухи під час перегляду різних об'єктів або дій [10], [12];

- засоби відстеження позиції та орієнтації. Програмне забезпечення OpenFaceTracker може визначати позицію та орієнтацію обличчя людини у просторі, що може бути корисним для додатків віртуальної або доповненої реальності [10], [12];

- дослідження соціальних взаємодій: програмне забезпечення OpenFaceTracker може допомагати аналізувати взаємодії між людьми на основі реакцій облич та емоцій [10], [12];

- засоби рекламної аналітики: в сфері маркетингу і реклами програмне забезпечення OpenFaceTracker може бути використано для аналізу реакцій споживачів на рекламні матеріали та продукти [10], [12];

- дослідження поведінки користувачів: програмне забезпечення OpenFaceTracker може використовуватися для дослідження психологічних та поведінкових аспектів людей в різних ситуаціях [10], [12];

- програмне забезпечення OpenFaceTracker може бути використано для керування інтерфейсами за допомогою рухів обличчя або очей [10], [12];

- специфічні маркери: програмне забезпечення OpenFaceTracker може розпізнавати специфічні маркери на обличчі, що дозволяє аналізувати різні аспекти обличчя [10], [12];

- підтримка різних операційних систем: програмне забезпечення OpenFaceTracker підтримує операційні системи Windows, Linux, OS X та Raspbian, що робить його доступним для різних платформ [10], [12];

- взаємодія з іншим програмним забезпеченням: програмне забезпечення OpenFaceTracker надає C++ API, що дозволяє розробникам вбудовувати його функціональність у власні програми [10], [12];

- вільна ліцензія: програмне забезпечення OpenFaceTracker поширюється під ліцензією Apache 2.0, що дозволяє вільне використання та модифікацію [10], [12];

- засоби аналізу соціальних взаємодій: програмне забезпечення OpenFaceTracker може використовуватися для аналізу соціальних взаємодій, реакцій на обличчя та емоції під час спілкування [10], [12];

- відкритий засіб: програмне забезпечення OpenFaceTracker базується на відкритих алгоритмах і може бути розширено та модифіковано за потребою користувачів або розробників [10], [12].

Загалом, програмне забезпечення OpenFaceTracker дозволяє використовувати технології відстеження обличчя для аналізу поведінки та взаємодії людей, що має застосування в різних галузях, включаючи науку, технології та маркетинг [10], [12].

Серед недоліків програмного забезпечення OpenFaceTracker можна відзначити те, що це програмне забезпечення є досить його складним в налаштування, що вимагає певних технічних знань для ефективного використання. Крім того, для ефективної роботи програмного забезпечення OpenFaceTracker може бути необхідний потужний апаратний комплекс, особливо при обробці великого обсягу зображень [10], [12].

Програмне забезпечення Betaface є професійною програмою, призначеною для локалізації та розпізнавання облич. Програмне забезпечення Betaface спеціально розроблене для використання медіа компаніями [13]. Це програмне забезпечення дозволяє визначати особи на фотографіях та отримувати інформацію про цих людей. Програмне забезпечення Betaface (рис. 1.3) має низку переваг, включаючи ідентифікацію широкого спектру деталей, таких як вік, вираз обличчя, наявність бороди або окулярів, кольори волосся, форму підборіддя, положення очей, форму та товщину брів,

довжину волосся, форму голови та інші дрібні характеристики. Також, програмний сервіс Betaface дозволяє знаходити схожих осіб у базі даних знаменитостей [13].

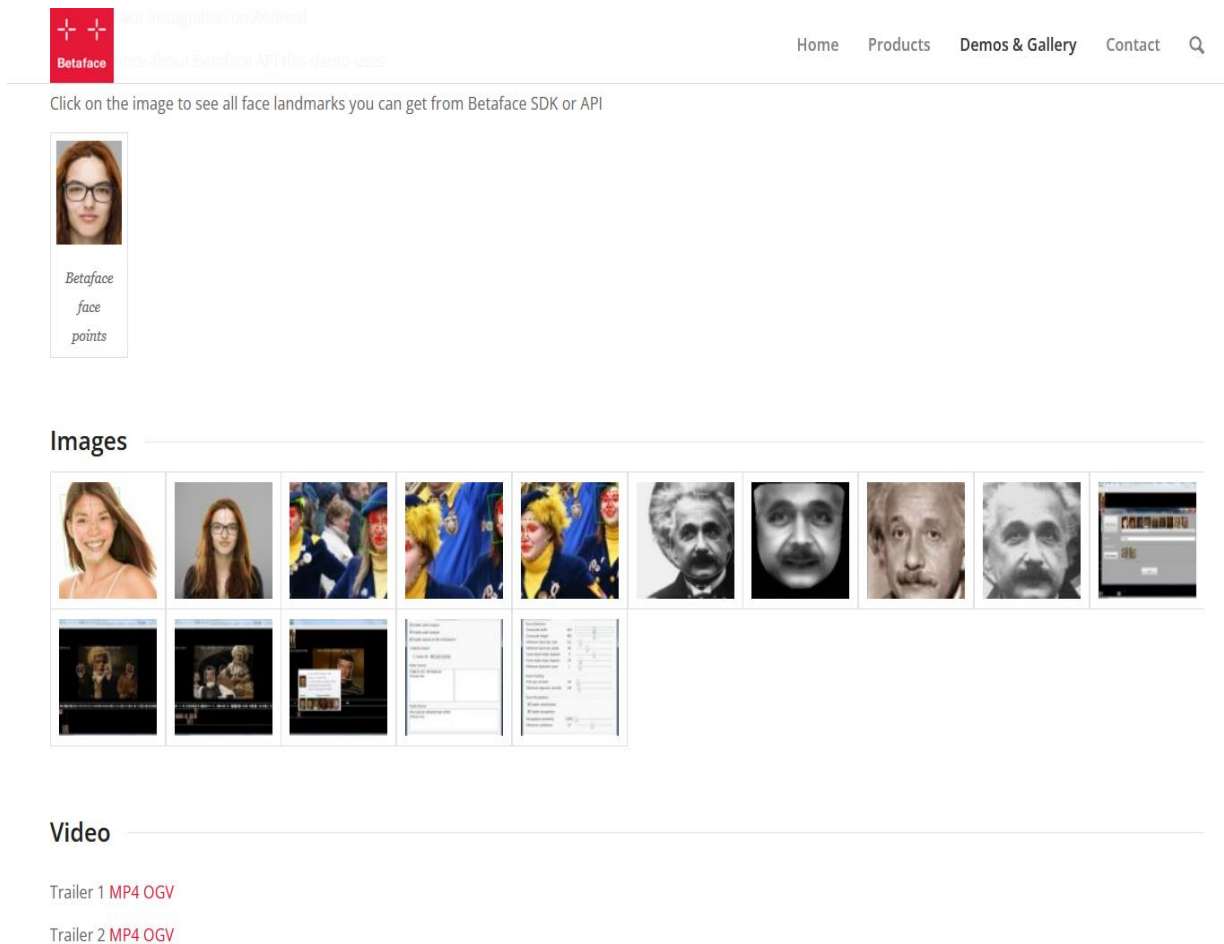


Рисунок 1.3 – Програмне забезпечення Betaface [13]

Програма Betaface призначена для розпізнавання облич і надає засоби для ідентифікації осіб на фотографіях [13]. Програма дозволяє автоматично аналізувати різні аспекти обличчя, такі як вираз, вік, наявність бороди або окулярів, кольори волосся, розташування очей, форма підборіддя, довжина волосся, форма голови та інші характеристики. Однією з основних функцій програми є здатність швидко і точно розпізнавати особи на фотографіях, що дозволяє визначити їхню ідентичність. Крім того, програма надає можливість знаходити схожі особи у базі даних знаменитостей, що може бути корисним для різних компаній та журналістів [13].

Серед особливостей програмного забезпечення Betaface для локалізації облич на зображеннях можна виділити такі [13]:

- можливість розпізнавання облич: програмне забезпечення Betaface спеціалізується на розпізнаванні облич і надає потужні інструменти для ідентифікації осіб на фотографіях [13];

- широкий спектр характеристик: програма Betaface надає детальну інформацію про різні аспекти обличчя, такі як вік, вираз обличчя, колір волосся, наявність окулярів чи бороди, розташування очей, форма підборіддя, товщина брів, форма голови тощо [13];

- швидке розпізнавання: програмне забезпечення Betaface дозволяє швидко розпізнавати обличчя на зображеннях, що є важливим для різних компаній та організацій, які працюють з великим обсягом зображень [13];

- можливість виявляти схожість між розпізнаними особами та знаменитостями з бази даних, що дозволяє знайти аналогії людини з відомими особами [13];

- простий інтерфейс: хоча програма орієнтована на професіоналів, вона надає зручний інтерфейс, що спрощує роботу з програмою [13];

- платна та безкоштовна версії: безкоштовна бета-версія дозволяє випробувати програму, але для повного функціоналу доступна платна версія програмного забезпечення Betaface з більш високою точністю розпізнавання та більшим набором можливостей [13];

- підтримка медіа компаній: програмне забезпечення Betaface спеціально розроблене для потреб медіа компаній, які працюють з великим обсягом зображень та потребують швидкого та точного розпізнавання облич різних персон [13].

Варто зазначити, що недоліком програмного забезпечення Betaface є те, що цей сервіс є платним. Однак існує безкоштовна бета-версія програми, яка має обмежений функціонал і меншу точність розпізнавання порівняно з платною версією [13].

Порівняльну характеристику програмного забезпечення для локалізації облич на зображеннях наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для локалізації облич на зображеннях

Критерій порівняння	OpenBR [11]	OpenFaceTracker [12]	Betaface [13]
Засоби інтеграції з іншим ПЗ	+	+-	-
Безкоштовність	+	+	+-
Засоби візуалізації даних	+-	+-	+
Зручність налаштувань	+-	-	+
Розширюваність	+	+-	+
Кросплатформність	+	+-	+-

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для локалізації облич на зображеннях. Проте впровадження та підтримка деяких програм для локалізації облич на зображеннях може вимагати значних фінансових і технічних витрат на навчання моделей, розробку та підтримку інфраструктури. У деяких випадках програм для локалізації облич на зображеннях можуть використовувати штучні обхідні методи для покращення точності, що може спотворити результати і робити їх непередбачуваними. Деякі програми можуть виявляти більш високу точність при розпізнаванні облич певних етнічних або культурних груп порівняно з іншими, що може призвести до некоректного розпізнавання або дискримінації. Тому актуальною є розробка програмного забезпечення для локалізації облич на зображеннях, що використовує різні методи виявлення та розпізнавання облич.

При розробці програмного забезпечення оптимізації для локалізації облич на зображеннях необхідно забезпечити такі функціональні вимоги:

- можливість завантаження фотографії для локалізації облич на ній;

- можливість локалізації обличчя (визначення меж обличчя);
- можливість взаємодії з базою даних для отримання обличчя певної персони;
- порівняння обличчя на фотографії з обличчям з бази даних;
- можливість визначення персони шляхом порівняння її фотографії з інформацією з бази даних;
- можливість додавання облич в базу даних;
- можливість введення інформацію про людину у базу даних (ім'я, фотографія, інша інформація), а також можливість редагування наявної інформації.

1.3 Висновки за розділом 1

Визначено, що локалізація облич є важливою частиною завдання розпізнавання облич та широко використовується у різних сферах людської діяльності, зокрема, в медицині, безпеки та відео спостережень, розваги та реклами. Проаналізовано підходи до локалізації облич на зображеннях. Виконано класифікацію методів локалізації облич. Досліджено етапи функціонування програмного забезпечення для локалізації облич на зображеннях. Описано використання методів та моделей машинного та глибокого навчання для локалізації облич. Проаналізовано сфери діяльності, де розпізнавання та локалізація облич допомагає спростити життя та забезпечити безпеку. Визначено переваги та недоліки використання програмних засобів для локалізації та розпізнавання облич. Проаналізовано проблеми використання методів локалізації та розпізнавання облич.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для локалізації облич на зображеннях. Проте впровадження та підтримка деяких програм для локалізації облич на зображеннях може вимагати значних фінансових і технічних витрат на навчання моделей, розробку та підтримку інфраструктури. У деяких випадках

програм для локалізації облич на зображеннях можуть використовувати штучні обхідні методи для покращення точності, що може спотворити результати і робити їх непередбачуваними. Деякі програми можуть виявляти більш високу точність при розпізнаванні облич певних етнічних або культурних груп порівняно з іншими, що може призвести до некоректного розпізнавання або дискримінації. Тому актуальною є розробка програмного забезпечення для локалізації облич на зображеннях, що використовує різні методи виявлення та розпізнавання облич.

Сформульовано функціональні вимоги до програмного забезпечення для локалізації облич на зображеннях.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення оптимізації для локалізації облич на зображеннях було використано мову програмування Python [14], [15].

Python – це мова програмування, яка поєднує в собі зрозумілість та потужність об'єктно-орієнтованого підходу, і може бути порівняна з іншими мовами, такими як Perl, Ruby та Java [14], [15].

Python є мовою програмування з високою інтерпретацією, вона розроблена для зручного читання коду. Як і в багатьох інших мовах, використані англійські ключові слова з пунктуацією, але у цій мові менше синтаксичних конструкцій, ніж в інших мовах [14], [15]. Python є мовою, що інтерпретується. Інтерпретатор обробляє код Python без потреби попереднього компілювання. Цей підхід подібний до того, як це відбувається в мовах Perl та PHP [14], [15].

Python є інтерактивним, дозволяючи взаємодіяти безпосередньо з інтерпретатором, навіть враховуючи підказки для написання та виконання власного коду [14], [15].

Підтримуючи об'єктно-орієнтований підхід, Python дозволяє використовувати об'єктно-орієнтовані стилі програмування, які дозволяють згрупувати код у об'єктах для зручності та ефективності [14], [15].

Мова програмування Python є чудовим вибором для розробки різного типу програмного забезпечення, оскільки ця мова дозволяє створювати програми різного рівня складності, від простої обробки тексту до розробки веббраузерів і навіть ігор [14], [15].

Основні переваги Python включають [14], [15]:

- зручний синтаксис, що робить код легко зрозумілим;
- простота використання, що спрощує процес програмування. Python добре підходить для створення прототипів і вирішення специфічних завдань без додаткових ускладнень;

- велика стандартна бібліотека, яка підтримує багато різноманітних завдань програмування, включаючи роботу з вебсерверами, обробку тексту за допомогою регулярних виразів та роботу з файлами;

- інтерактивний режим дозволяє швидко тестувати короткі фрагменти коду. Також існує пакетне середовище розробки;

- легкість розширення за допомогою додаткових модулів, написаних на інших мовах програмування, таких як C або C++;

- можливість вбудовування Python в інші програми для створення програмованих інтерфейсів;

- кросплатформеність: Python працює на різних операційних системах. Крім того, існують неофіційні збірки для платформ Android та iOS [14], [15].

Ця мова програмування є безкоштовною, її можна змінювати та поширювати без обмежень, оскільки ця мова захищена авторським правом, але діє під ліцензією з відкритим кодом [14], [15].

Деякі особливості мови програмування Python такі [14], [15]:

- велика кількість основних типів даних, таких як числа (включаючи числа з плаваючою комою, комплексні числа та довгі цілі без обмежень), рядки (ASCII та Unicode), а також списки та словники;

- Python підтримує об'єктно-орієнтоване програмування, включаючи можливість створювати класи та використовувати успадкування;

- код на Python може бути організованим у модулі та пакети для зручності;

- мова Python надає можливість створювати та обробляти винятки для більш чистої обробки помилок;

- типи даних в Python є динамічними та динамічно типізованими. Використання несумісних типів (наприклад, рядків та чисел) спричиняє винятки, що допомагає вчасно виявляти помилки;

- зручний синтаксис, який нагадує англійську мову та робить код читабельним і зрозумілим;

- велика спільноту розробників, яка підтримує багато різноманітних бібліотек та модулів;
- Python надає розширені функції програмування, такі як генератори та розуміння списків;
- мова Python автоматично керує пам'яттю, усуваючи потребу вручну виділяти та звільняти пам'ять у вашому коді [14], [15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для локалізації облич на зображеннях наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для локалізації облич на зображеннях

Критерій порівняння мов програмування	Мова програмування		
	Python	Perl	Ruby
Об'єктно-орієнтованість	+	+—	+
Зручність для розробки програмного забезпечення для локалізації облич на зображеннях	+	—	—
Підтримка розробників	+	+—	+—
Розширення для завдань обробки зображень	+	+—	+—
Продуктивність	+	+—	+
Розширюваність	+	+—	+

Отже, для реалізації програмного забезпечення для локалізації облич на зображеннях обрано мову програмування Python, яка є безкоштовною, зручною, широко використовуваною для створення програмного забезпечення різного призначення, а також має багато модулів для створення програм для оброблення зображень.

2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для локалізації облич на зображеннях

В якості середовища розробки для створення програмного забезпечення для локалізації облич на зображеннях було обрано PyCharm [16], [17].

PyCharm є одним з ефективних інтегрованих середовищ розробки для мови програмування Python. Це пояснюється кількома факторами, зокрема тим, що його розробила компанія JetBrains, відома своєю популярною IDE IntelliJ IDEA, яка є однією з кращих для Java, та WebStorm, яка вважається найрозумнішою JavaScript IDE. Також, важливим аспектом є підтримка веброзробки з використанням Django [16], [17].

Кілька факторів роблять PyCharm одним з найбільш повних та комплексних інтегрованих середовищ розробки для роботи з мовою програмування Python [16], [17].

PyCharm є інтегрованим середовищем розробки для мови програмування Python, яке пропонується як багатоплатформенний застосунок, сумісний з різними операційними системами, зокрема, Linux, macOS і Window [16], [17].

PyCharm постачається з багатьма модулями, пакетами та інструментами, які прискорюють процес розробки на Python, значно зменшуючи зусилля, потрібні для виконання рутинних завдань. Крім того, програму можна налаштувати відповідно до особистих уподобань та вимог проекту [16], [17].

PyCharm надає також такі можливості, як графічний редактор для відлагодження коду, інтегрований блок-тестер для тестування програм, а також інтеграцію з системами контролю версій для зручного керування версіями розроблюваного коду [16], [17].

Як додаткова можливість, для роботи з науковими даними PyCharm може бути використано разом з Anaconda, що дозволяє ефективно працювати

над розробленням програмного забезпечення для аналізу даних та відповідних програмних інструментів [16], [17].

PyCharm був розроблений з основною метою спростити програмування на мові Python і забезпечити зручну роботу на різних операційних системах, таких як Windows, Linux і macOS. Це інтегроване середовище розробки включає в себе різноманітні інструменти для аналізу коду, налагодження, тестування та управління версіями. Крім цього, PyCharm дозволяє розробникам створювати власні модулі для Python, використовуючи доступні інтерфейсні засоби [16], [17].

Це інтегроване середовище розробки спрощує роботу з різними базами даних, надаючи можливість взаємодіяти з ними без необхідності використання інших інструментів. Незважаючи на те, що PyCharm спеціально спрямований на роботу з Python, в ньому також можна створювати файли типу HTML, CSS і Javascript. Інтерфейс користувача дуже зручний і приємний, і його можна налаштувати за допомогою різноманітних шаблонів [16], [17].

Однією з суттєвих переваг PyCharm є його розумний редактор коду, який полегшує процес написання якісного коду на Python. Він виділяє ключові слова, класи та функції спеціальними кольорами, що робить код більш зрозумілим та читабельним. Також це середовище розробки надає можливість підсвічування помилок, що сприяє виявленню та виправленню неточностей у програмному коді [16], [17].

PyCharm надає суттєву підтримку при розробці на мові програмування Python, включаючи сумісність з різними версіями мови, відлагодження коду та інтеграцію з віртуальними середовищами [16], [17].

PyCharm відповідає за керування науковими бібліотеками, такими як Matplotlib та SciPy, для допомоги розробникам Python у втіленні проєктів, пов'язаних з обробкою даних та машинним навчанням [16], [17].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для локалізації облич на зображеннях наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для локалізації облич на зображеннях

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Visual Studio Code	Eclipse
Мови програмування	Переважно Python	Різні мови, включаючи Python	Відкритий для різних мов
Засоби аналізу даних та машинного навчання	+	+—	+—
Модулі для обробки зображень	+	—	+—
Інтелектуальне завершення коду	+	+—	+—
Зручні інструменти для тестування ПЗ	+	+	—

Таким чином, для створення програмного забезпечення для локалізації облич на зображеннях обрано середовище розробки PyCharm, що постачається з багатьма модулями, пакетами та інструментами, які прискорюють процес розробки на Python, значно зменшуючи зусилля, потрібні для виконання рутинних завдань при створенні програм обробки зображень.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях

З метою можливості розробки програмного забезпечення для локалізації облич на зображеннях створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях враховуємо те, що користувачі програми мають можливості завантаження фотографії для локалізації облич на ній, локалізації обличчя (визначення меж обличчя), взаємодії з базою даних для отримання обличчя певної особи, порівняння обличчя на фотографії з обличчям з бази даних, визначення особи шляхом порівняння її фотографії з інформацією з бази даних, додавання облич в базу даних, введення інформацію про людину у базу даних (ім'я, фотографія, інша інформація), а також можливість редагування наявної інформації.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для локалізації облич на зображеннях можна описати таким чином:

- користувач програмного забезпечення для локалізації облич на зображеннях запускає програму;
- після запуску програми локалізації облич на зображеннях користувачу відображається головна форма;
- користувач програми локалізації облич на зображеннях бачить головну форму та завантажує зображення для пошуку обличчя;
- користувач здійснює локалізації облич на зображеннях шляхом натискання відповідної кнопки у програмі;
- користувач за необхідності додає фото нової особи у базу даних;
- користувач завершує роботу з програмним забезпеченням для локалізації облич на зображеннях.

Враховуючи сценарії роботи користувачів при роботі з програмним забезпеченням для локалізації облич на зображеннях, модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

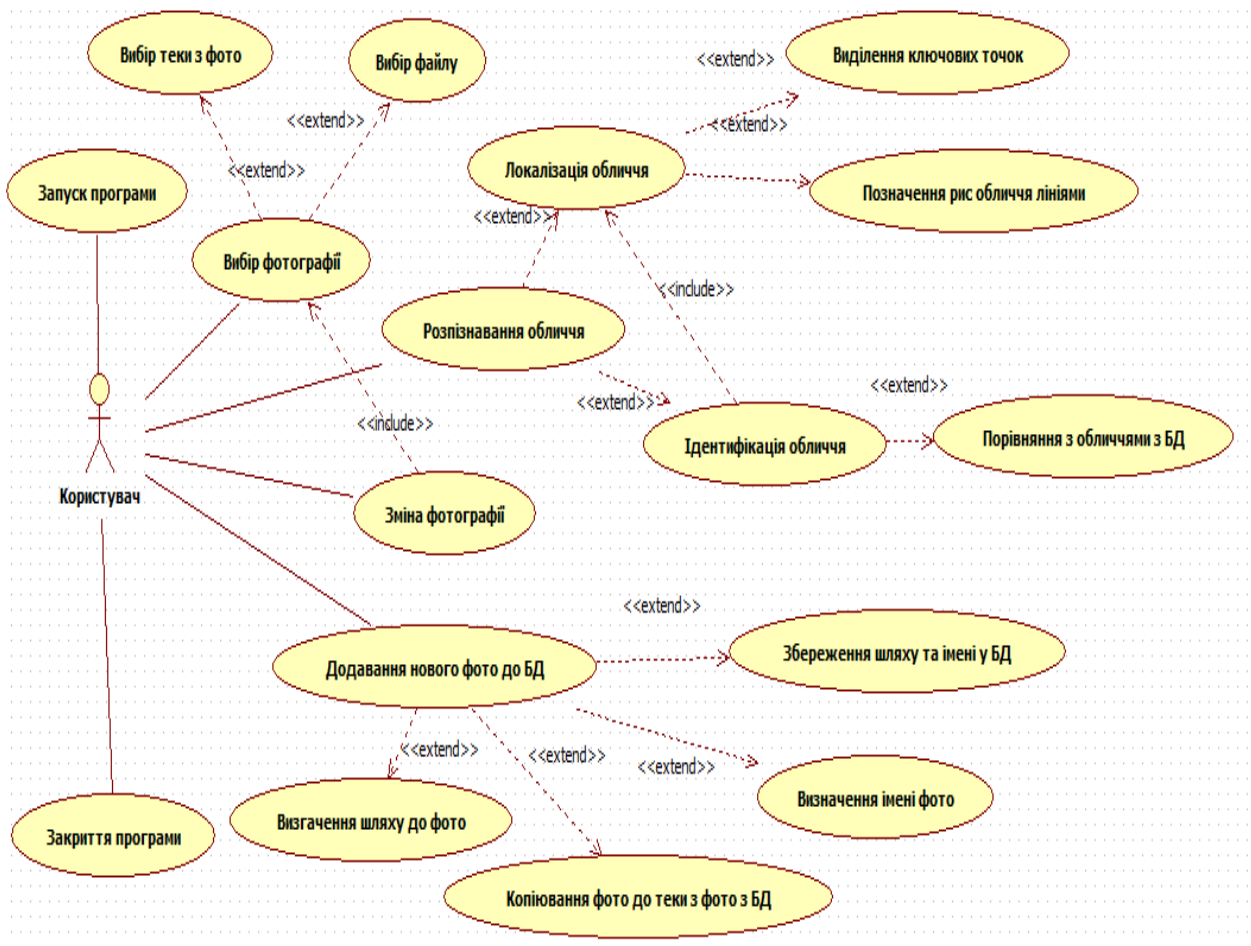


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для локалізації облич на зображеннях.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для локалізації облич на зображеннях обрано мову програмування Python, яка є безкоштовною, зручною, широко використовуваною для створення програмного забезпечення різного призначення, а також має багато модулів для створення програм для оброблення зображень.

Для створення програмного забезпечення для локалізації облич на зображеннях обрано середовище розробки PyCharm, що постачається з багатьма модулями, пакетами та інструментами, які прискорюють процес розробки на Python, значно зменшуючи зусилля, потрібні для виконання рутинних завдань при створенні програм обробки зображень.

Створено модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для локалізації облич на зображеннях.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для локалізації облич на зображеннях

Структуру програмного забезпечення для локалізації облич на зображеннях наведено на рис. 3.1.

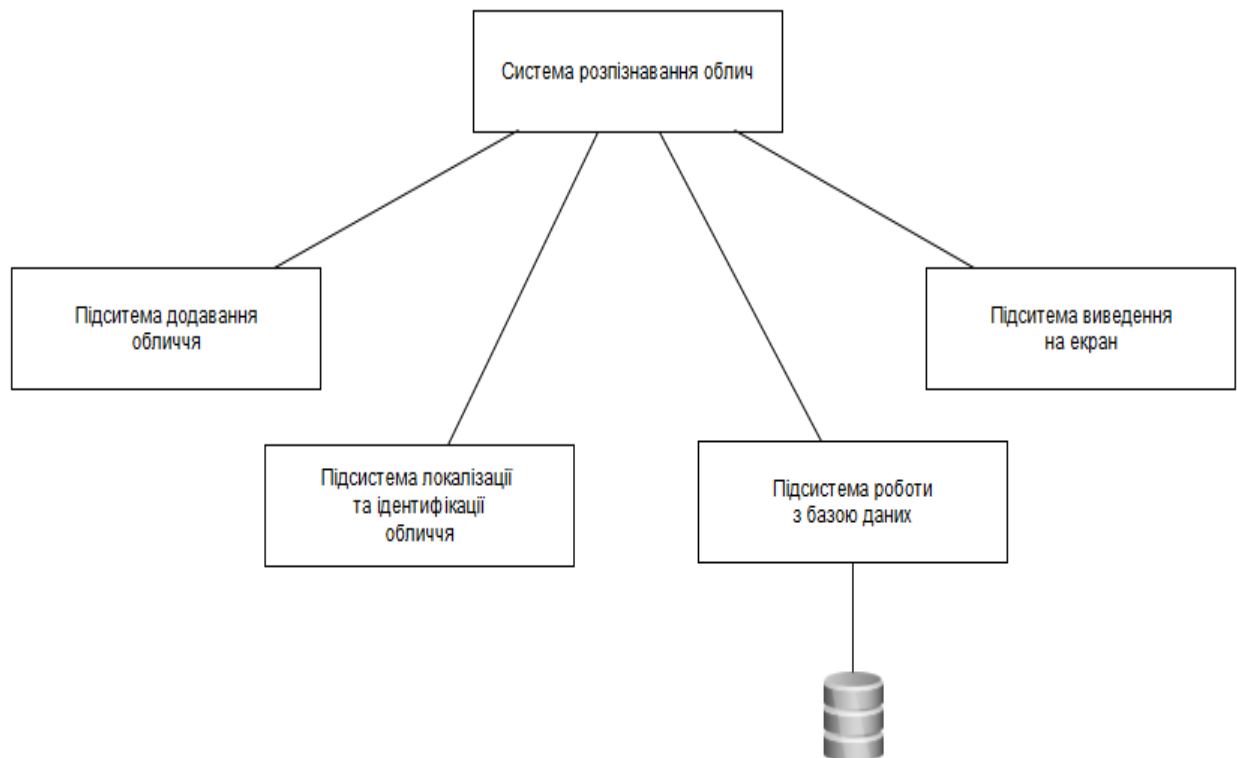


Рисунок 3.1 – Структура програмного забезпечення для локалізації облич на зображеннях

Як видно з рисунку, програмне забезпечення для локалізації облич на зображеннях складається з таких компонентів:

- підсистема додавання облич;
- підсистема локалізації та ідентифікації обличчя;
- підсистема роботи з базою даних наявних облич;
- підсистема виведення на екран результатів роботи програмного забезпечення для локалізації облич на зображеннях.

Проаналізуємо призначення основних компонентів програмного забезпечення для локалізації облич на зображеннях.

Підсистема локалізації та ідентифікації облич забезпечує знаходження обличчя на фотографії (або повідомляє про його відсутність), виділення меж обличчя та його частин (очей, носу, рота), порівняння обличчя з наявними у базі, визначення імені зображуваної особи.

Підсистема додавання обличчя забезпечує можливість поповнення бази облич новими екземплярами. Якщо система не змогла розпізнати обличчя, дана підсистема запропонує користувачу і надасть необхідні інструменти для додавання обличчя до бази.

Підсистема роботи з базою даних виконує всі операції, пов'язані з додаванням або отриманням інформації з бази даних.

Підсистема виведення на екран забезпечує візуальний інтерфейс системи, надає можливість користувачу виводу інформації на екран, а також отримання даних, введених користувачем.

Розроблюване програмне забезпечення для локалізації облич на зображеннях складається з наступних модулів: Program, MainForm, Face, AddForm і DataBase.

Модуль Program містить клас Program, який складає основу програми, бо з нього викликаються функції реалізовані у інших модулях. У класі program реалізовано наступні методи:

- locallizationFace() – завантажує обране користувачем зображення, обробляє його за допомогою методів, реалізованих у модулі recognitionFunctions, виділяє основні риси обличчя (форму, очі, ніс, рот) і зафарбовує їх. Таким чином здійснюється локалізація обличчя на фотографії.

- identificationPerson() – функція на основі фото, завантажених з бази даних, та за допомогою методів, реалізованих у модулі recognitionFunctions, здійснює ідентифікацію персони на фотографії, визначає її ім'я.

- restartProgram() – функція перезапускає програму.

– `addPhoto()` – функція викликається при додаванні до бази даних нового зображення. Вона копіює картинку з поточної теки до теки зі всіма фото з БД, визначає шлях до файлу та викликає функцію для занесення інформації в БД.

Модуль `MainForm` містить клас `MainForm`, який відповідає за графічний інтерфейс головної форми програми. Клас реалізує наступні функції:

– `initUI()` – у функції описано елементи головної форми, які відображаються при запуску програми, їх розташування у формі і команди, які виконуються при взаємодії з цими елементами.

– `main()` – функція описує вигляд (розмір, розташування та ін.) головної форми.

– `label()` – виводить напис у головну форму програми.

– `myFile()` – відкриває файловий менеджер, дозволяє обрати зображення з бажаної теки та завантажити його.

– `close()` – закриває програму.

Модуль `Face` описує клас `Face`, який містить реалізацію всіх функцій, необхідних для локалізації та ідентифікації персони. У модулі наявні наступні функції:

– `_rect_to_css()` – перетворює об'єкт бібліотеки `dlib` «`rect`» на кортеж;

– `_css_to_rect()` – перетворює кортеж у об'єкт «`rect`» бібліотеки `dlib`;

– `_trim_css_to_bounds()` – перевіряє, чи знаходиться кортеж у межах зображення;

– `face_distance()` – приймає список кодування обличчя, порівнює їх з відомими кодуваннями облич, розраховує евклідову відстань для кожного порівняння;

– `load_image_file()` – завантажує файл із зображенням у масив;

– `face_locations()` – повертає масив з контуром обмеження меж обличчя;

– `face_landmarks()` – повертає масив з описом обличчя (розташування очей, носа, рота);

- face_encodings() – повертає масив з кодом обличчя;
- compare_faces() – порівнює список кодування двох облич.

Модуль AddForm описує клас AddForm і містить функції, які забезпечують графічний інтерфейс для додавання фотографії до бази даних.

Перелік функцій:

- main() – функція забезпечує головні налаштування форми;
- showAddForm() – функція виводить поле для написання імені зображеної на фото людини та кнопку, яка обробляє введену інформацію;
- show_message() – показує інформаційне повідомлення користувачу.

Модуль DataBase забезпечує роботу з базою даних. Модуль описує клас Database, який містить наступні методи:

- selectAll() – повертає масив, який містить шлях до кожного зображення;
- numberPhoto() – повертає кількість фотографій у базі даних;
- getName() – повертає ім'я людини, зображеної на фотографії;
- addPhoto() – додає фото у базу даних.

Програмне забезпечення для локалізації облич на зображеннях при розпізнаванні обличчя розв'язує декілька взаємопов'язаних задач:

- знаходження обличчя на зображенні;
- розпізнання облич, навіть якщо вони схилені або якщо погане освітлення;
- виокремлення унікальних рис обличчя, які відрізняють одну людину від іншої (наприклад, розмір очей, форма обличчя та ін.);
- порівняння унікальних особливостей обличчя з усіма людьми, яких система вже знає, для ідентифікації.

На кожному з цих етапів використовуються різні алгоритми розпізнавання образів, зокрема, алгоритм пошуку гістограми спрямованих градієнтів, алгоритм оцінки орієнтирів обличчя, згортова нейронна мережа, метод опорних векторів.

3.2 Функціонування програмного забезпечення для локалізації облич на зображеннях

Схема функціонування програмного забезпечення для локалізації облич на зображеннях наведена на рис. 3.2.

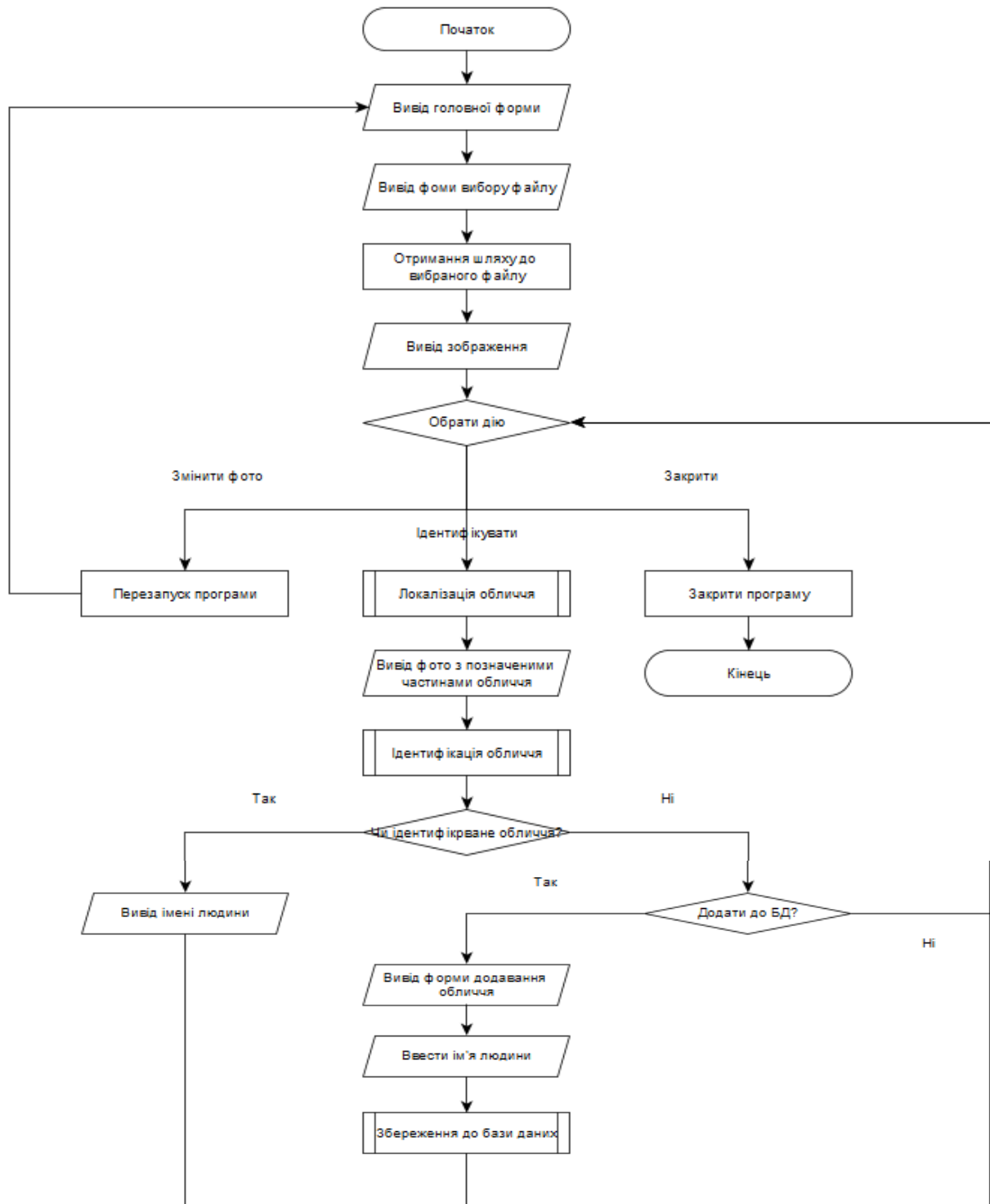


Рисунок 3.2 – Схема функціонування програмного забезпечення для локалізації облич на зображеннях

Як видно з рисунку 3.2, після запуску програми користувач вводить зображення для локалізації обличчя, потім виводиться обране користувачем зображення. Після цього користувач обирає дію (виконання локалізації обличчя, перезапуск або вихід з програми). Далі відбувається розпізнавання обличчя, внаслідок чого виводиться фотографія з позначеними частинами обличчя. Після цього виконується ідентифікація обличчя. У користувача є можливість зберегти нове обличчя та інформацію про відповідну персону у базі даних.

3.3 Проєктування програмного забезпечення для локалізації облич на зображеннях

Загальна діаграма класів програмного забезпечення для локалізації облич на зображеннях наведена на рис. 3.3.

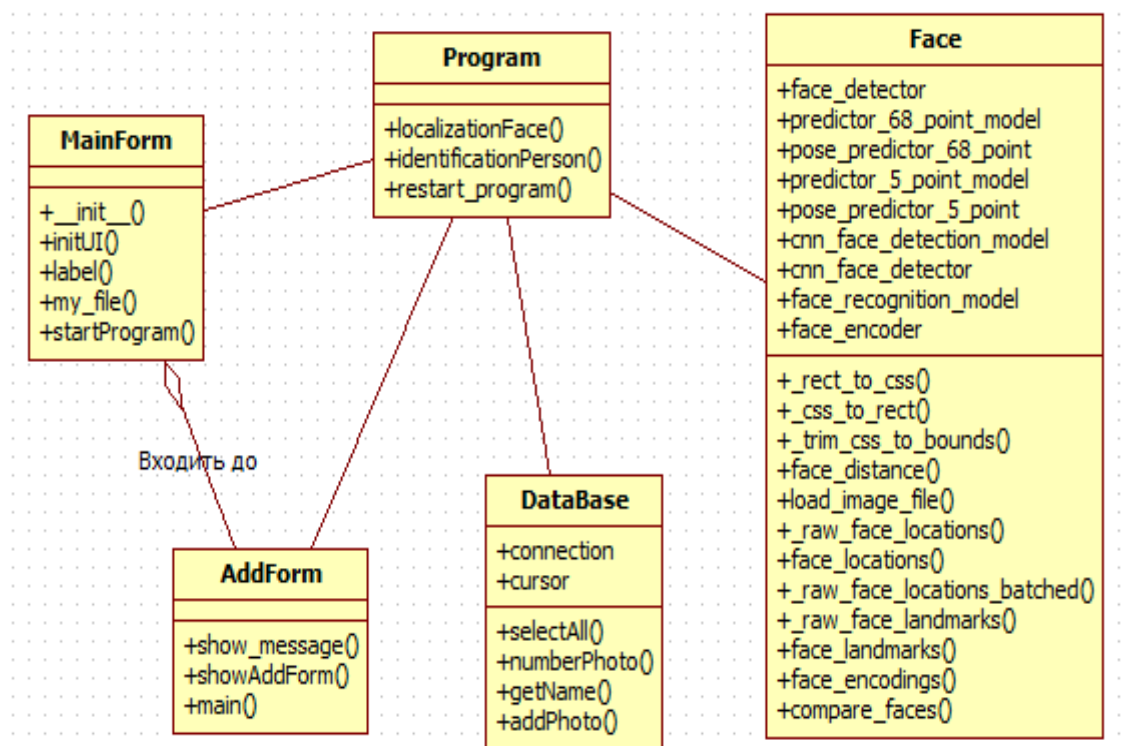


Рисунок 3.3 – Діаграма класів програмного забезпечення для локалізації облич на зображеннях

Як видно з діаграми на рис. 3.3, основними класами програмного забезпечення для локалізації облич на зображеннях є такі: головна форма (MainForm), програма (Program), обличчя (Face) та інші.

Діаграма компонентів програмного забезпечення для локалізації облич на зображеннях наведена на рисунку 3.4. Дана діаграма описує особливості фізичної уяви системи.

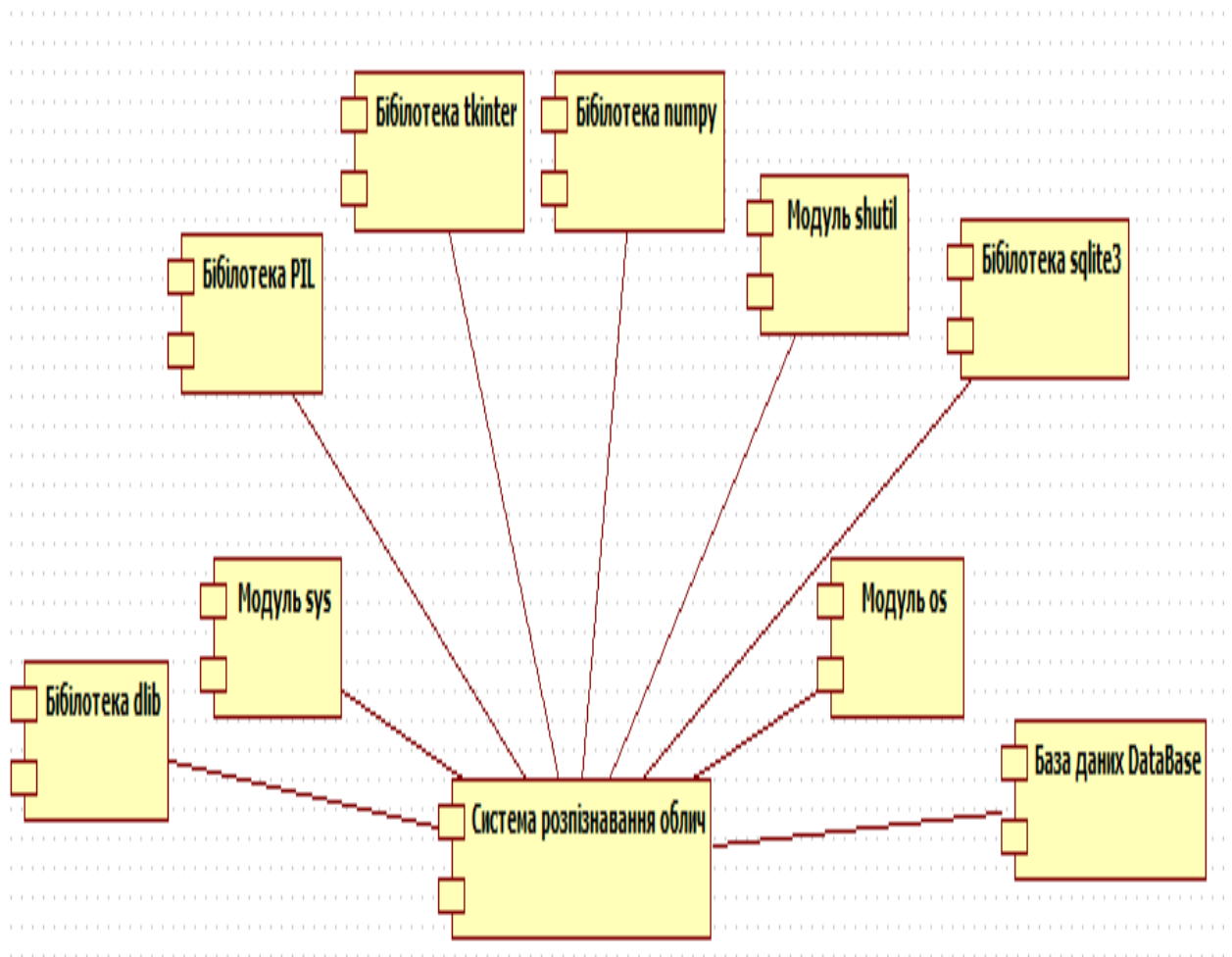


Рисунок 3.4 – Діаграма компонентів програмного забезпечення для локалізації облич на зображеннях

Діаграма кооперацій програмного забезпечення для локалізації облич на зображеннях наведена на рисунку 3.5.

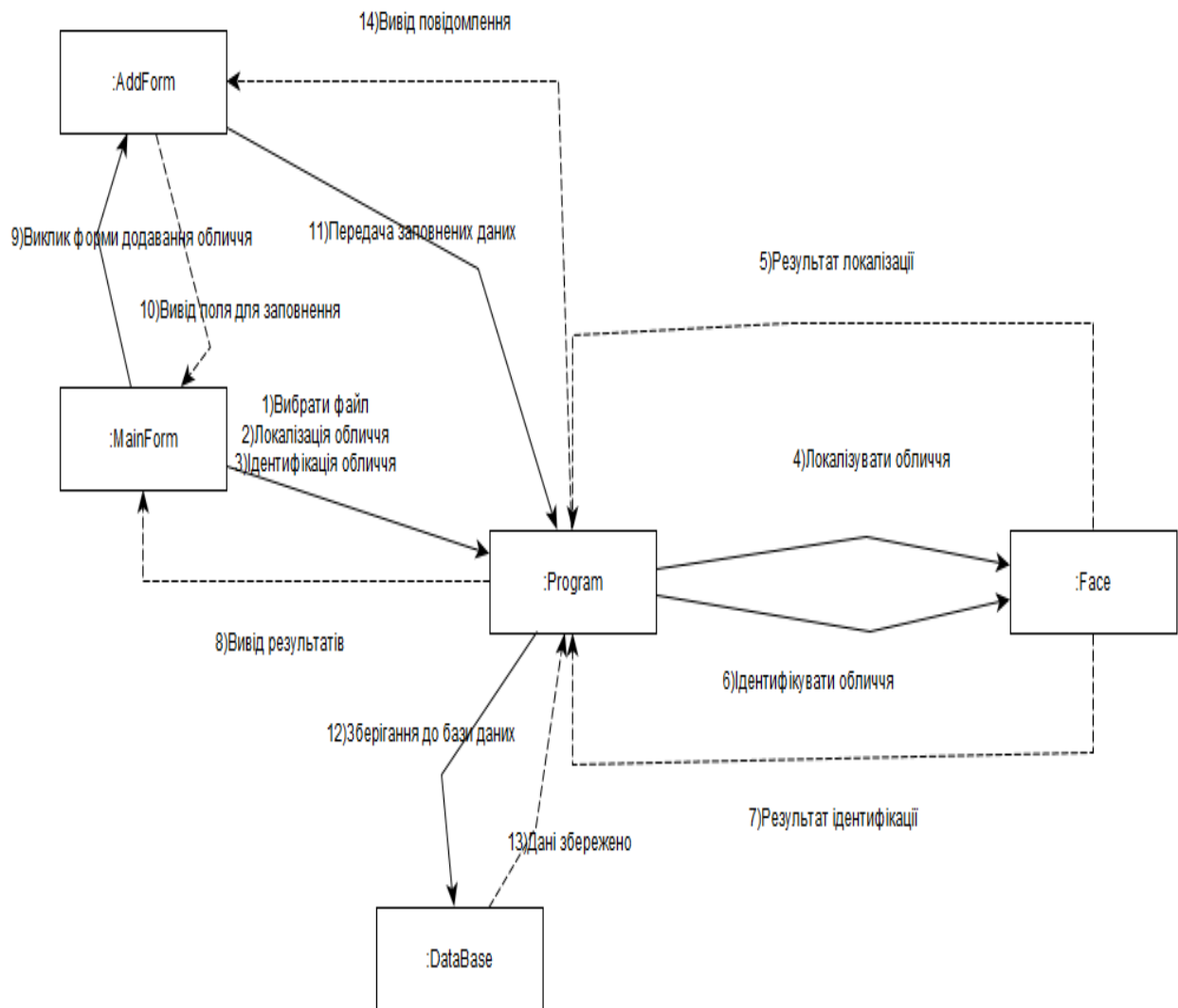


Рисунок 3.5 – Діаграма кооперацій програмного забезпечення для локалізації облич на зображеннях

3.4 Проєктування інтерфейсу програмного забезпечення для локалізації облич на зображеннях

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях виконано з урахуванням вимог технічного завдання.

Приклади розроблених інтерфейсних форм наведено на рис. 3.6-3.10.

Форму головного вікна програмного забезпечення для локалізації облич на зображеннях з обраним фото для розпізнавання наведено на рис. 3.6.



Рисунок 3.6 – Форма головного вікна програми з обраним фото для розпізнавання

Форму діалогового вікна вибору фотографій для розпізнавання обличч відображено на рис. 3.7.

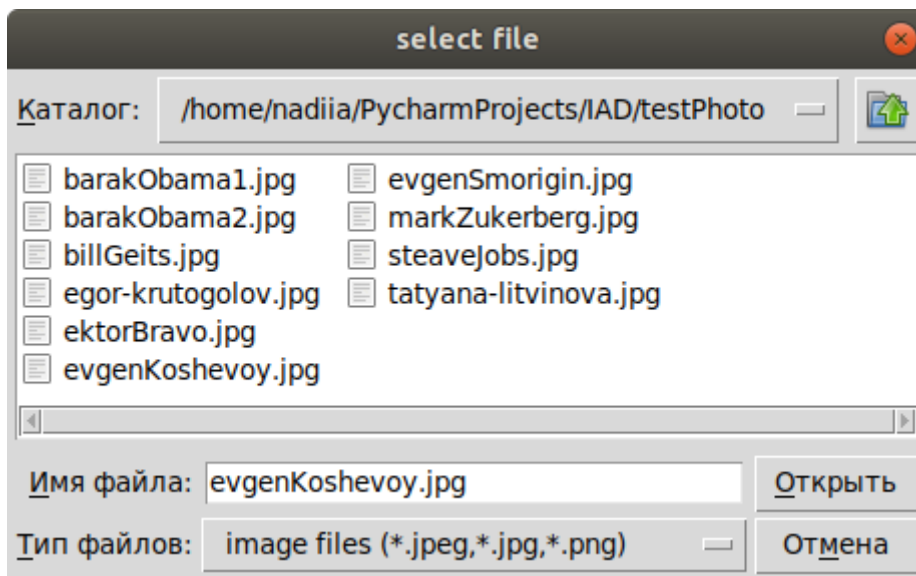


Рисунок 3.7 – Форма діалогового вікна вибору фотографій для розпізнавання обличч

Форму вікна з результатами локалізації обличчя з позначенням рис обличчя людини (підборіддя, ніс, очі, рот, брови) наведено на рис. 3.8.



Рисунок 3.8 – Форма вікна з результатами локалізації обличчя з позначенням рис обличчя людини

Інтерфейс форми додавання фотографії нової персони до бази даних програми відображено на рис. 3.9.



Рисунок 3.9 – Інтерфейс форми додавання фотографії нової персони до бази даних програми

Діалогове вікно з повідомленням про успішне додавання нової фотографії до бази даних наведено на рис. 3.10.

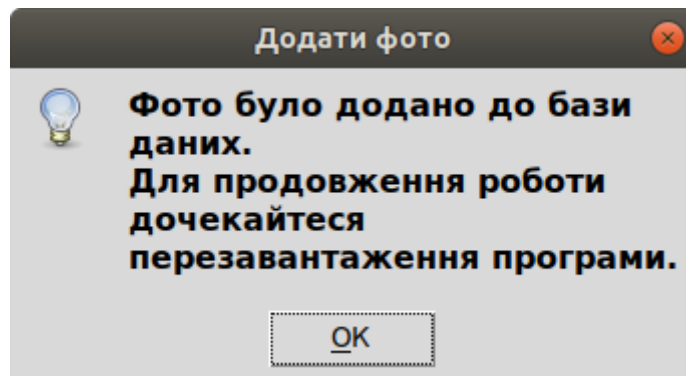


Рисунок 3.10 – Діалогове вікно з повідомленням про успішне додавання нової фотографії до бази даних

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для локалізації облич на зображеннях. Запропоновано структуру та описано основні модулі програмного забезпечення для локалізації облич на зображеннях. Визначено, що основними компонентами програми для локалізації облич на зображеннях є: підсистема додавання облич, підсистема локалізації та ідентифікації обличчя, підсистема роботи з базою даних наявних облич, підсистема виведення на екран результатів роботи.

Описано функціонування програмного забезпечення для локалізації облич на зображеннях. Описано особливості реалізації програмного забезпечення для локалізації облич на зображеннях. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для локалізації облич на зображеннях. Програма написана на мові Python, яка є безкоштовною, зручною, широко використовуваною для створення програмного забезпечення різного призначення, а також має багато модулів для створення програм для оброблення зображень. В якості середовища розробки для створення програми обрано PyCharm, що постачається з багатьма модулями, пакетами та інструментами, які прискорюють процес розробки на Python, значно зменшуючи зусилля, потрібні для виконання рутинних завдань при створенні програм обробки зображень.

Програмне забезпечення для локалізації облич на зображеннях забезпечує виконання таких функцій:

- можливість завантаження фотографії для локалізації облич на ній;
- можливість локалізації обличчя (визначення меж обличчя);
- можливість взаємодії з базою даних для отримання обличчя певної персони;
- порівняння обличчя на фотографії з обличчям з бази даних;
- можливість визначення персони шляхом порівняння її фотографії з інформацією з бази даних;
- можливість додавання облич в базу даних;
- можливість введення інформацію про людину у базу даних (ім'я, фотографія, інша інформація), а також можливість редагування наявної інформації.

Для експлуатації програмного забезпечення для локалізації облич на зображеннях необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 2,5 GHz і більше;
- вільне місце на диску 500 Мб або більше;

- оперативна пам'ять об'ємом 4Гб і більше;
- інтерпретатор мови програмування Python версії, не нижче за 3.

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

4.2 Характеристики програми для локалізації облич на зображеннях

Програмне забезпечення для локалізації облич на зображеннях може працювати безперервно, в залежності від потреб користувача.

Програмне забезпечення для локалізації облич на зображеннях може використовуватись в будь-якій операційній системі, де встановлено інтерпретатор Python версії не нижче третьої.

Для коректного функціонування програми для локалізації облич на зображеннях вміст директорії з програмним забезпеченням не можна змінювати.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для локалізації облич на зображеннях необхідно виконати python скрипт mainrozr.pyw, що знаходиться в кореневій директорії програми.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для локалізації облич на зображеннях є зображення, що обираються користувачем для

розпізнавання, а також інформація про нову персону, яку програма не може ідентифікувати на основі наявних даних.

Вихідними даними програмного забезпечення для локалізації облич на зображеннях є результат розпізнавання зображення, зокрема зображення з обличчям із позначенням рис (підборіддя, ніс, очі, рот, брови), а також повідомлення про успішне додавання нової фотографії до бази даних.

4.3.3 Повідомлення

Користувач програмного забезпечення для локалізації облич на зображеннях може отримати різні повідомлення, зокрема, повідомлення про успішне додавання нової фотографії до бази даних.

4.4 Виконання програмного забезпечення для локалізації облич на зображеннях

Після запуску програмного забезпечення для локалізації облич на зображеннях користувачу виводиться головна форма програми (рис. 4.1).

Форму головного вікна програмного забезпечення з обраним фото для розпізнавання наведено на рис. 4.1. Як видно, користувач має можливість завантажити або змінити фото для розпізнавання обличчя, а також ідентифікувати персону шляхом порівняння її фото із тим, що зберігається у базі дані.

При натисканні на кнопку Змінити фото з'являється форма діалогового вікна вибору файлу з зображенням (рис. 4.2.). Тут користувач може обрати файл із зображенням, локалізацію обличчя якого необхідно виконати.

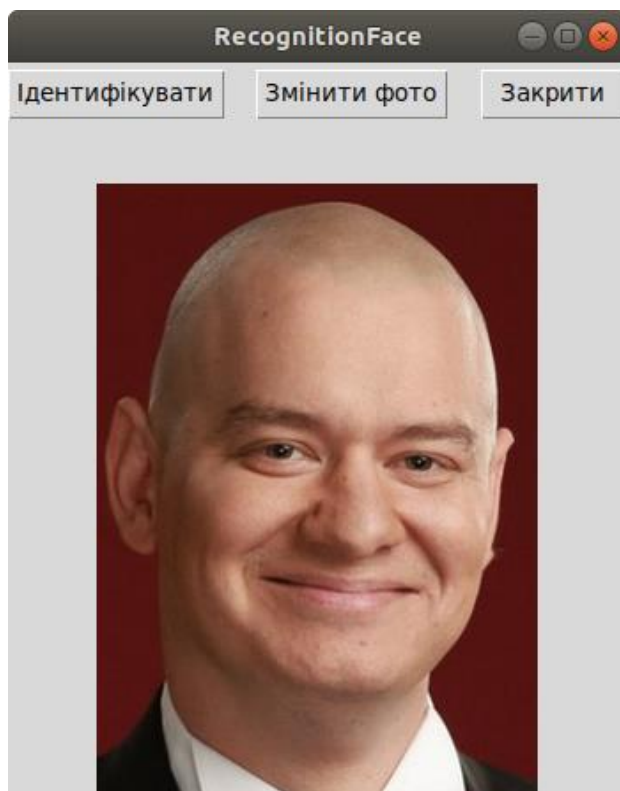


Рисунок 4.1 – Форма головного вікна програми з обраним фото для розпізнавання

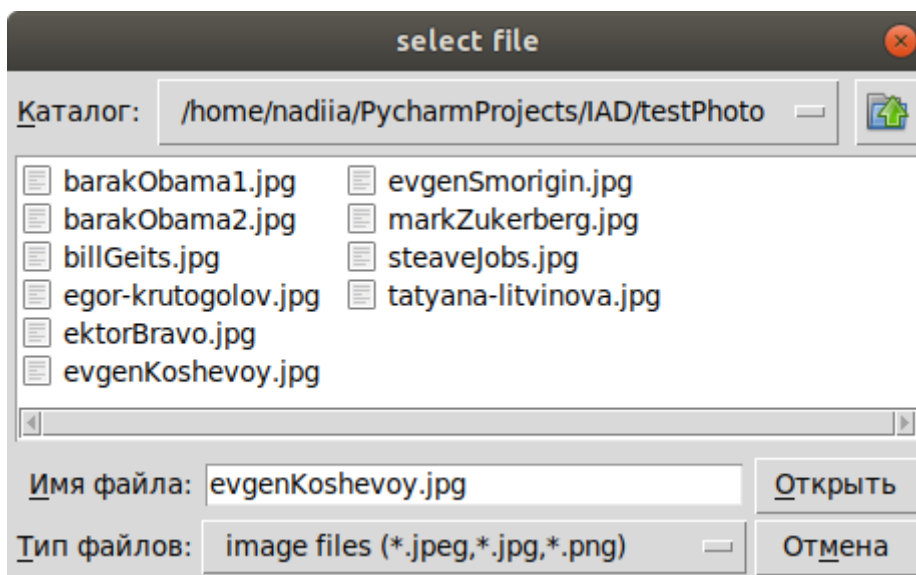


Рисунок 4.2 – Форма діалогового вікна вибору зображення для локалізації обличчя

Як результат локалізації виводиться вікно зображенням обличчя з позначенням рис обличчя людини (підборіддя, ніс, очі, рот, брови) (рис. 4.3).



Рисунок 4.3 – Форма вікна з результатами локалізації обличчя з позначенням рис обличчя людини

У випадку, якщо персони при розпізнаванні не знайдено у базі даних, користувач має можливість додати відповідну інформацію у базу даних. Інтерфейс форми додавання фотографії нової персони до бази даних програми відображено на рис. 4.4.



Рисунок 4.4 – Інтерфейс форми додавання фотографії нової персони до бази даних програми

Після додавання нової фотографії до бази даних виводиться відповідне діалогове вікно.

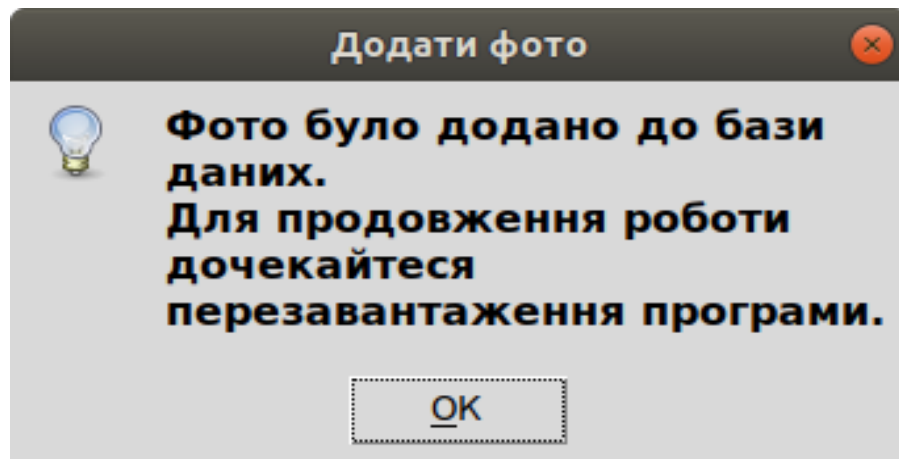


Рисунок 4.5 – Форма вікна з повідомленням про успішне додавання нової фотографії до бази даних

4.5 Тестування програмного забезпечення для локалізації облич на зображеннях

Виконано тестування програмного забезпечення для локалізації облич на зображеннях.

Виявлено, що програма для локалізації облич на зображеннях функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу – локалізацію облич на зображеннях.

Приклад успішного розпізнавання зображень за допомогою даного програмного забезпечення наведено на рисунках 4.6 та 4.7.

На роботу програми не вплинули навіть погане освітлення обличчя, а також нахил обличчя. Наприклад, на рис.4.7 зображено нахилене донизу обличчя, але програма усе одно правильно розпізнає зображену людину.

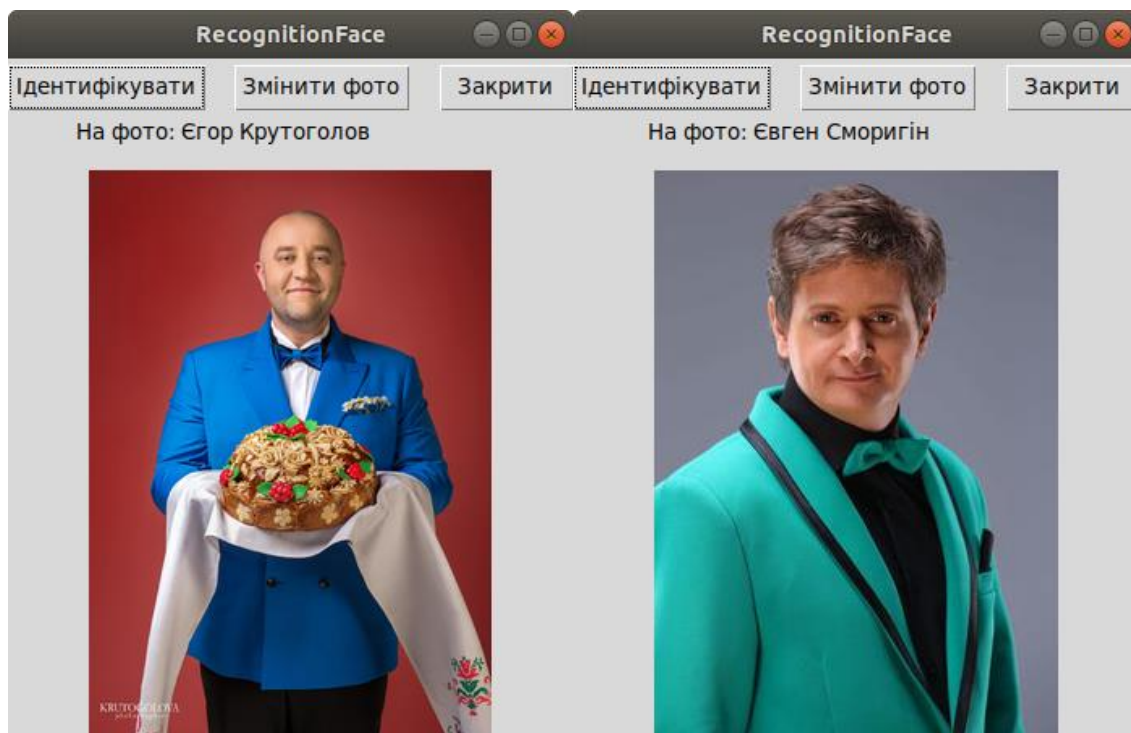


Рисунок 4.6 – Приклад успішного розпізнавання зображень за допомогою даного програмного забезпечення



Рисунок 4.7 – Приклад успішної локалізації та ідентифікації нахиленого обличчя

Результати проведеного успішного тестування за допомогою модульних тестів наведено на рис.4.8.

▼ ✓ Test Results	7 s 157 ms
▼ ✓ UnitTest	7 s 157 ms
▼ ✓ Test_DataBase	2 ms
✓ test_getName	1 ms
✓ test_numberPhoto	0 ms
✓ test_selectAll	1 ms
▼ ✓ Test_Face	902 ms
✓ test_compare_face	702 ms
✓ test_face_landmar	187 ms
✓ test_load_image_fil	13 ms
▼ ✓ Test_Program	6 s 253 ms
✓ test_identificationPerson	

Рисунок 4.8 – Результати успішного тестування за допомогою модульних тестів

Отже, розроблене програмне забезпечення дозволяє розв’язувати завдання локалізації та розпізнавання зображень.

4.6 Висновки за розділом 4

Описано програмне забезпечення для локалізації облич на зображеннях. Виконано тестування розробленого програмного забезпечення для локалізації облич на зображеннях. Результати тестування програмного забезпечення для локалізації облич на зображеннях показали, що розроблена програма дозволяє розв’язувати завдання розпізнавання облич.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для обробки зображень.

Визначено, що локалізація облич є важливою частиною завдання розпізнавання облич та широко використовується у різних сферах людської діяльності, зокрема, в медицині, безпеки та відео спостережень, розваги та реклами. Проаналізовано підходи до локалізації облич на зображеннях. Виконано класифікацію методів локалізації облич. Досліджено етапи функціонування програмного забезпечення для локалізації облич на зображеннях. Описано використання методів та моделей машинного та глибокого навчання для локалізації облич. Проаналізовано сфери діяльності, де розпізнавання та локалізація облич допомагає спростити життя та забезпечити безпеку. Визначено переваги та недоліки використання програмних засобів для локалізації та розпізнавання облич. Проаналізовано проблеми використання методів локалізації та розпізнавання облич.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для локалізації облич на зображеннях. Проте впровадження та підтримка деяких програм для локалізації облич на зображеннях може вимагати значних фінансових і технічних витрат на навчання моделей, розробку та підтримку інфраструктури. У деяких випадках програм для локалізації облич на зображеннях можуть використовувати штучні обхідні методи для покращення точності, що може спотворити результати і робити їх непередбачуваними. Деякі програми можуть виявляти більш високу точність при розпізнаванні облич певних етнічних або культурних груп порівняно з іншими, що може призвести до некоректного розпізнавання або дискримінації. Тому актуальною є розробка програмного забезпечення для локалізації облич на зображеннях, що використовує різні методи виявлення та розпізнавання облич.

Сформульовано функціональні вимоги до програмного забезпечення для локалізації облич на зображеннях.

Для реалізації програмного забезпечення для локалізації облич на зображеннях обрано мову програмування Python, яка є безкоштовною, зручною, широко використовуваною для створення програмного забезпечення різного призначення, а також має багато модулів для створення програм для оброблення зображень.

Для створення програмного забезпечення для локалізації облич на зображеннях обрано середовище розробки PyCharm, що постачається з багатьма модулями, пакетами та інструментами, які прискорюють процес розробки на Python, значно зменшуючи зусилля, потрібні для виконання рутинних завдань при створенні програм обробки зображень.

Новизна роботи полягає в тому, що запропоновано модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для локалізації облич на зображеннях.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для локалізації облич на зображеннях.

Запропоновано структуру та описано основні модулі програмного забезпечення для локалізації облич на зображеннях. Визначено, що основними компонентами програми для локалізації облич на зображеннях є: підсистема додавання облич, підсистема локалізації та ідентифікації обличчя, підсистема роботи з базою даних наявних облич, підсистема виведення на екран результатів роботи.

Описано функціонування програмного забезпечення для локалізації облич на зображеннях. Описано особливості реалізації програмного забезпечення для локалізації облич на зображеннях. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях.

Виконано тестування розробленого програмного забезпечення для локалізації облич на зображеннях. Результати тестування програмного забезпечення для локалізації облич на зображеннях показали, що розроблена програма дозволяє розв'язувати завдання розпізнавання облич.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Face Detection For Beginners [Electronic resource]. – Access mode : <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>.
2. Machine learning and face recognition [Electronic resource]. – Access mode : <https://www.pxl-vision.com/en/blog/machine-learning-and-how-it-applies-to-facial-recognition-technology>.
3. What is facial recognition and how does it work [Electronic resource]. – Access mode : <https://us.norton.com/blog/iot/how-facial-recognition-software-works>.
4. Face detection [Electronic resource]. – Access mode : <https://www.techtarget.com/searchenterpriseai/definition/face-detection>.
5. Facial recognition: top 7 trends (tech, vendors, use cases) [Electronic resource]. – Access mode : <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>.
6. Face Detection: Real-Time Deep Learning Applications [Electronic resource]. – Access mode : <https://viso.ai/deep-learning/face-detection-overview/>.
7. What is Face Detection? Ultimate Guide 2023 [Electronic resource]. – Access mode : <https://learnopencv.com/what-is-face-detection-the-ultimate-guide/>.
8. 5 Best Face Detection Software that will Gain Prominence in 2023 [Electronic resource]. – Access mode : <https://www.analyticsinsight.net/5-best-face-detection-software-that-will-gain-prominence-in-2023/>.
9. Facial Recognition Software: Top Vendors in 2023 [Electronic resource]. – Access mode : <https://www.techrepublic.com/article/facial-recognition-software/>.
10. The Best 3 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. OpenBR [Electronic resource]. – Access mode :
<https://www.goodfirms.co/face-detection-software/blog/best-free-open-source-face-detection-software-solutions>.

12. OpenFaceTracker [Electronic resource]. – Access mode :
<https://openfacetracker-lib3.readthedocs.io/en/latest/>.

13. Betaface [Electronic resource]. – Access mode :
<https://www.betaface.com/wpa/>.

14. Python [Electronic resource]. – Access mode :
<https://wiki.python.org/moin/BeginnersGuide/Overview>.

15. Python - Overview [Electronic resource]. – Access mode :
https://www.tutorialspoint.com/python/python_overview.htm.

16. PyCharm. Overview [Electronic resource]. – Access mode :
<https://blog.jetbrains.com/pycharm/2020/12/pycharm-2020-3-overview/>.

17. What is PyCharm? Features, Advantages & Disadvantages [Electronic resource]. – Access mode : <https://hackr.io/blog/what-is-pycharm>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для локалізації облич на зображеннях.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів локалізації облич на зображеннях», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

A.2 Призначення розробки

Програмний продукт призначений для забезпечення та підтримки процесу локалізації облич на зображеннях.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для локалізації облич на зображеннях забезпечує виконання таких функцій:

- можливість завантаження фотографії для локалізації облич на ній;
- можливість локалізації обличчя (визначення меж обличчя);
- можливість взаємодії з базою даних для отримання обличчя певної персони;
- порівняння обличчя на фотографії з обличчям з бази даних;
- можливість визначення персони шляхом порівняння її фотографії з інформацією з бази даних;
- можливість додавання облич в базу даних;

– можливість введення інформацію про людину у базу даних (ім'я, фотографія, інша інформація), а також можливість редагування наявної інформації.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для локалізації облич на зображеннях повинен бути зручним для користувачів.

А.3.3 Вимоги до надійності

Програмне забезпечення для локалізації облич на зображеннях повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для локалізації облич на зображеннях необхідна наявність персонального комп'ютера у користувача.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для локалізації облич на зображеннях необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 2,5 GHz і більше;
- вільне місце на диску 500 Мб або більше;
- оперативна пам'ять об'ємом 4Гб і більше;
- інтерпретатор мови програмування Python версії, не нижче за 3.

А.3.6 Вимоги до маркування і пакування

Програма для локалізації облич на зображеннях може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для локалізації облич на зображеннях».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для локалізації облич на зображеннях є зображення, що обираються користувачем для розпізнавання, а також інформація про нову персону, яку програма не може ідентифікувати на основі наявних даних.

Вихідними даними програмного забезпечення для локалізації облич на зображеннях є результат розпізнавання зображення, зокрема зображення з обличчям із позначенням рис (підборіддя, ніс, очі, рот, брови), а також повідомлення про успішне додавання нової фотографії до бази даних.

ДОДАТОК Б
Фрагмент тексту програми

```

class adFrm:
    def shwmsg(self):
        messagebox.showinfo("Додати фото", "Фото було додано
до бази даних.\nДля продовження роботи дочекайтеся
перезавантаження програми.")
    def showAdFrm(self, self2):
        fnmn = Prg.fnmn
        global txt
        txt = Entry(self2)
        txt.grid(ryad=3, column=0, sticky=W + E)
        kbtn = Button(self2, txt="OK", command=lambda:
[self.shwmsg(), Prg.addPhoto(fnmn, txt.get()),
prg.rstrtart_prg()])
        kbtn.grid(ryad=3, column=2)
    def mnbv(self, root, self2):
        bbtn = Button(self2, txt="Додати\ndo BD",
command=lambda:[self.showAdFrm(self2)])
        bbtn.grid(ryad=2, column=2)
        root.mnbvloop()

class Dbs:
    cnctn = zapytite3.connect("Dbs")
    crsr = cnctn.cursor()
    def slctl(self):
        zapyt = "SELECT shlh FROM Photo"
        klk = 0
        rslps = [0] * self.nmbPht()[0]
        for ryad in self.crsr.execute(sql):
            rslps[klk] = ryad
            klk += 1

```

```

        print(rslps)
        return rslps
    def nmbPht(self):
        zapyt = "SELECT klk(*) FROM Photo"
        self.cursor.execute(sql)
        klk = self.cursor.fetchone()
        print(klk)
        return klk
    def gtnm(self, shlhToFile):
        zapyt = "SELECT nmmf FROM Photo WHERE shlh=?"
        self.cursor.execute(sql, [(shlhToFile)])
        nmmf = self.cursor.fetchone()
        print(nmmf)
        return nmmf
    def addPhoto(self, shlh, nmmf):
        params = [shlh, nmmf]
        print(shlh)
        print(nmmf)
        zapyt = """INSERT INTO Photo(shlh, nmmf) VALUES(?,
?)"""
        self.cursor.execute(sql, params)
        self.cnctn.commit()

class Ovlych:
    ovlychdtct = dlib.get_frontal_ovlychdtct()
    pr68mod = ovlych_rcgnt_mdls.pspr_md1_lctnd()
    psprdct68pnt = dlib.shape_predictor(pr68mod)
    pr5pntmdl =
ovlych_rcgnt_mdls.pspr_five_point_md1_lctnd()
    pspr5mdl = dlib.shape_predictor(pr5pntmdl)
    cnn_ovlych_dtctn_md1 =
ovlych_rcgnt_mdls.cnn_ovlychdtct_md1_lctnd()

```

```

        cnn_ovlychdtct =
dlib.cnn_ovlych_dtctn_md1_v1(cnn_ovlych_dtctn_md1)
        ovlych_rcgnt_md1 =
ovlych_rcgnt_mdls.ovlych_rcgnt_md1_lctnd()
        ovlych_ncdf = dlib.ovlych_rcgnt_md1_v1(ovlych_rcgnt_md1)
        def ovlych_distance(self, ovlych_ncdngs,
ovlych_to_compare):
            if len(ovlych_ncdngs) == 0:
                return np.empty((0))
            return np.linalg.norm(ovlych_ncdngs -
ovlych_to_compare, axis=1)
        def lifm(self, file, mode='RGB'):
            ifm = PIL.Image.open(file)
            if mode:
                ifm = ifm.convert(mode)
            return np.asarray(ifm)
        def _raw_ovlych_lctnds(self, imag, ntud=1, mdl="hog"):
            if mdl == "cnn":
                return self.cnn_ovlychdtct(imag, ntud)
            else:
                return self.ovlychdtct(imag, ntud)
        def ovlych_lctnds(self, imag, ntud=1, mdl="hog"):
            if mdl == "cnn":
                return
[self._trim_css_to_bounds(self._rect_to_css(ovlych.rect),
imag.shape) for ovlych in self._raw_ovlych_lctnds(imag,
ntud, "cnn")]
            else:
                return
[self._trim_css_to_bounds(self._rect_to_css(ovlych),
imag.shape) for ovlych in self._raw_ovlych_lctnds(imag,
ntud, mdl)]

```

```

    def _raw_ovlych_lctnds_batched(self, images, ntud=1,
batch_size=128):
        return self.cnn_ovlychdtct(images, ntud,
batch_size=batch_size)
    def _raw_ovlych_lndmtrs(self, ovlych_image,
ovlych_lctnds=None, mdl="large"):
        if ovlych_lctnds is None:
            ovlych_lctnds =
self._raw_ovlych_lctnds(ovlych_image)
        else:
            ovlych_lctnds = [self._css_to_rect(ovlych_lctnd)
for ovlych_lctnd in ovlych_lctnds]
            pspr = self.psprdct68pnt
            if mdl == "small":
                pspr = self.pspr5mdl
            return [pspr(ovlych_image, ovlych_lctnd) for
ovlych_lctnd in ovlych_lctnds]
    def ovlych_lndmtrs(self, ovlych_image,
ovlych_lctnds=None, mdl="large"):
        lndmtrs = self._raw_ovlych_lndmtrs(ovlych_image,
ovlych_lctnds, mdl)
        lndmtrs_as_tuples = [(p.x, p.y) for p in
lndmtr.parts()] for lndmtr in lndmtrs]
        if mdl == 'large':
            return [{
                "chn": ptns[0:17],
                "lftyebryad": ptns[17:22],
                "rtyebryad": ptns[22:27],
                "nssbr": ptns[27:31],
                "nss_tip": ptns[31:36],
                "lftye": ptns[36:42],
                "rtye": ptns[42:48],

```

```

        "tplp": ptns[48:55] + [ptns[64]] +
[ptns[63]] + [ptns[62]] + [ptns[61]] + [ptns[60]],
        "btmlp": ptns[54:60] + [ptns[48]] +
[ptns[60]] + [ptns[67]] + [ptns[66]] + [ptns[65]] +
[ptns[64]]

        } for ptns in lndmtrs_as_tuples]
    elif mdl == 'small':
        return [{
            "nss_tip": [ptns[4]],
            "lftype": ptns[2:4],
            "rtype": ptns[0:2],
        } for ptns in lndmtrs_as_tuples]
    else:
        raise ValueError("Помилки ['small', 'large'].")
    def ovlych_ncdngs(self, ovlych_image,
known_ovlych_lctnds=None, num_jitters=1):
        """
        raw_lndmtrs = self._raw_ovlych_lndmtrs(ovlych_image,
known_ovlych_lctnds, mdl="small")
        return
[ np.masvay(self.ovlych_ncdf.compute_ovlych_descriptor(ovlych
_image, raw_lndmtr_set, num_jitters)) for raw_lndmtr_set in
raw_lndmtrs ]
        def compare_ovlychs(self, known_ovlych_ncdngs,
ovlych_ncdng_to_check, tlrc=0.6):
            return
list(self.ovlych_distance(known_ovlych_ncdngs,
ovlych_ncdng_to_check) <= tlrc)

class MnbfvForm(Frame):
    def mfl(self):

```

```

        Prg.fnmn =
filedialog.askopenfnmn(initialdir="/home/dp/tstph",

title="select file", filetypes=(
        ("файли", " *.jpeg"), ("файли", " *.jpg"),
("файли", " *.png")))

    def startPrg(self, root):
        global rstr
        prg.lklzOvlych()
        rstr = prg.idfpr(Prg.fnmn)
        if (rstr == ""):
            rstr = "Не можу ідентифікувати. \nРаніше не
бачив цю людину."
            self.label()
            adFrm.mnbv(root, self)
            self.label()
            print(rstr)
def mnbv():
    global rtt
    rtt = Tk()
    rtt.geometry("350x420+300+300")
    apt = MnbvForm(rtt)
    apt.initUI()
    cnvs = Canvas(rtt, width=250, height=350)
    img = Image.open(Prg.fnmn)
    img = image.rstrize((250, 350), Image.ANTIALIAS)
    photo = ImageTk.PhotoImage(img)
    cnvs.create_image(0, 0, anchor=NW, img=photo)
    cnvs.pack()
    rtt.mnbvloop()
if __nmmf__ == '__mnbv__':
    mnbv()

```

```

class Prg:
    def lklzOvlych(self):
        shlh_to_file = fnmn
        img = ovlych.lifm(shlh_to_file)
        ovlych_lndmtrs_list = ovlych.ovlych_lndmtrs(img)
        print("Я знайшов {} обличчя на цій
фотографії.".format(len(ovlych_lndmtrs_list)))
        pil_img = Img.frommasvay(img)
        d = ImgDraw.Draw(pil_img)
        for ovlych_lndmtrs in ovlych_lndmtrs_list:
            for fcfl in ovlych_lndmtrs.keys():
                print("{} на цьому обличчі має координати:
{}".format(fcfl, ovlych_lndmtrs[fcfl]))
                for fcfl in ovlych_lndmtrs.keys():
                    d.line(ovlych_lndmtrs[fcfl], width=5)
        pil_img.show()
    def idfpr(self, shlh_to_file):
        nmbPht = dbs.nmbPht()
        klk = 100000
        masv = [0] * klk
        masv[0] = []
        vsepht = np.masvay(nmbPht)
        vsepht = dbs.slctl()
        i = 0
        masv2 = [0] * nmbPht[0]
        knwnovlychs = [0] * nmbPht[0]
        print(vsepht)
        while(i < len(vsepht)):
            img = vsepht[i][0]
            masv[i] = ovlych.lifm(img)
            uknwimg = ovlych.lifm(shlh_to_file)

```

```

        try:
            masv2[i] = ovlych.ovlych_ncdngs(masv[i])[0]
            uknwovlych_ncdng =
ovlych.ovlych_ncdngs(uknwimg)[0]
            knwnovlychs[i] = masv2[i]
            i += 1
            rslps = ovlych.compare_ovlychs(knwnovlychs,
uknwovlych_ncdng)
            i = 0
            klk = len(rslps)
            print(klk)
            while (True):
                if (rslps[i] == True):
                    global nmmf
                    nmmf = dbs.gtnm(vsepht[i][0])[0]
                    nmmf = "Ha photo: " + nmmf
                    print(nmmf)
                    break
                elif (i == (klk - 1)):
                    nmmf = ""
                    break
                else:
                    i = i + 1
            return nmmf
def rsttart_prg(self):
    pthn = sys.executable
    os.execl(pthn, pthn, * sys.argv)
def addPhoto(fnmn, nmmf):
    print(fnmn)
    photoNmmf = fnmn.split("/")
    shlh = "/home/dp/img/" + photoNmmf[-1]
    shtl.copy2(fnmn, shlh)
    dbs.addPhoto(shlh, nmmf)

```

```

class Test_Dbs(unittest.TestCase):
    def test_nmbPht(self):
        self.assertEqual(dbs.nmbPht(), (9,))
    def test_gtnm(self):
        shlh = "/home/dp/img/egor-krutogolov.jpg"
        self.assertEqual(dbs.gtnm(shlh), ("Crop
Крутоголов",))
    def test_slctl(self):
        zapyt = "SELECT shlh FROM Photo"
        klk = 0
        rslps = [0] * dbs.nmbPht()[0]
        for ryad in dbs.cursor.execute(sql):
            rslps[klk] = ryad
            klk += 1
        self.assertEqual(dbs.slctl(), rslps)
class Test_Prg(unittest.TestCase):
    def test_idfpr(self):
        shlh = "/home/dp/img/bill-gejts.jpg"
        self.assertEqual(Prg.idfpr(shlh), "На фото: Біл
Гейтс")
class Test_Ovlych(unittest.TestCase):
    def test_ovlych_lndmtrs(self):
        mdl = "large"
        ovlych_lctnds = None
        ovlych_img = Ovlych.lifm("/home/dp/img/bill-
gejts.jpg")
        lndmtrs = Ovlych._raw_ovlych_lndmtrs(ovlych_img,
ovlych_lctnds, mdl)
        lndmtrs_as_tuples = [(p.x, p.y) for p in
lndmtr.parts()] for lndmtr in lndmtrs]
        masvay = [{

```

```

        "chn": ptns[0:17],
        "lftyebryad": ptns[17:22],
        "rtyebryad": ptns[22:27],
        "nssbr": ptns[27:31],
        "nss_tip": ptns[31:36],
        "lftye": ptns[36:42],
        "rtye": ptns[42:48],
        "tplp": ptns[48:55] + [ptns[64]] + [ptns[63]] +
[ptns[62]] + [ptns[61]] + [ptns[60]],
        "btmlp": ptns[54:60] + [ptns[48]] + [ptns[60]] +
[ptns[67]] + [ptns[66]] + [ptns[65]] + [ptns[64]]
    } for ptns in lndmtrs_as_tuples]
    self.assertEqual(Ovlych.ovlych_lndmtrs(ovlych_img),
masvay)

    def test_compare_ovlychs(self):
        load = Ovlych.lifm("/home/dp/img/bill-gejts.jpg")
        known_ovlych_ncdngs = Ovlych.ovlych_ncdngs(load)[0]
        load2 =
Ovlych.lifm("/home/dp/testPhoto/billGeits.jpg")
        ovlych_ncdng_to_check = Ovlych.ovlych_ncdngs(load2)
        tlrc = 0.6
        masv =
list(Ovlych.ovlych_distance(known_ovlych_ncdngs,
ovlych_ncdng_to_check) <= tlrc)

self.assertEqual(Ovlych.compare_ovlychs(known_ovlych_ncdngs,
ovlych_ncdng_to_check, tlrc), masv)
if __nmmf__ == '__mnbv__':
    unittest.mnbv()

```

ДОДАТОК В
Слайди презентації

Дослідження та програмна реалізація засобів локалізації облич на зображеннях

Виконав: студент групи КНТ-112м

Семен КРИВА

Керівник: д.т.н., професор

Андрій ОЛІЙНИК

Рисунок В.1 – Слайд 1

2

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процес розробки програмного забезпечення для обробки зображень.
- **Предмет дослідження** – програмні засоби для локалізації облич на зображеннях.
- **Мета роботи** – дослідження та розробка програмного забезпечення для локалізації облич на зображеннях

Рисунок В.2 – Слайд 2

Завдання роботи

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

- виконати аналіз предметної області та програмних засобів для обробки зображень шляхом їх сегментації;
- розробити модель взаємодії користувача з програмним забезпеченням для сегментації зображень;
- здійснити проектування програмного забезпечення для сегментації зображень;
- створити програмне забезпечення для сегментації зображень;
- виконати тестування розробленого програмного забезпечення для сегментації зображень.

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для локалізації облич на зображеннях

Критерій порівняння	OpenBR	OpenFace Tracker	Betaface
Засоби інтеграції з іншим ПЗ	+	+-	-
Безкоштовність	+	+	+-
Засоби візуалізації даних	+-	+-	+
Зручність налаштувань	+-	-	+
Розширюваність	+	+-	+
Кросплатформність	+	+-	+-

Рисунок В.4 – Слайд 4

Обґрунтування вибору мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	Perl	Ruby
Об'єктно-орієнтованість	+	+-	+
Зручність для розробки програмного забезпечення для локалізації обличч на зображеннях	+	-	-
Підтримка розробників	+	+-	+-
Розширення для завдань обробки зображень	+	+-	+-
Продуктивність	+	+-	+
Розширюваність	+	+-	+

Рисунок В.5 – Слайд 5

Обґрунтування вибору середовища розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Visual Studio Code	Eclipse
Мови програмування	Переважно Python	Різні мови, включаючи Python	Відкритий для різних мов
Засоби аналізу даних та машинного навчання	+	+-	+-
Модулі для обробки зображень	+	-	+-
Інтелектуальне завершення коду	+	+-	+-
Зручні інструменти для тестування ПЗ	+	+	-

Рисунок В.6 – Слайд 6

```

    usecaseDiagram
        actor User as Користувач
        usecase UC1 as Запуск програми
        usecase UC2 as Вибір фотогалереї
        usecase UC3 as Вибір теги з фото
        usecase UC4 as Вибір файлу
        usecase UC5 as Локалізація обличчя
        usecase UC6 as Розпізнавання обличчя
        usecase UC7 as Зніма фотографії
        usecase UC8 as Ідентифікація обличчя
        usecase UC9 as Порівняння з обличчями з БД
        usecase UC10 as Додавання нового фото до БД
        usecase UC11 as Збереження шляху та імені у БД
        usecase UC12 as Визначення шляху до фото
        usecase UC13 as Висношення імені фото
        usecase UC14 as Копіювання фото до теги з фото з БД
        usecase UC15 as Позначення рис обличчя лініями
        usecase UC16 as Виділення ключових точок

        User -- UC1
        User -- UC2
        User -- UC3
        User -- UC4
        User -- UC5
        User -- UC6
        User -- UC7
        User -- UC10
        User -- UC12
        User -- UC14

        UC2 -.-> UC3 : <<extend>>
        UC2 -.-> UC4 : <<extend>>
        UC2 -.-> UC5 : <<extend>>
        UC2 -.-> UC6 : <<include>>
        UC2 -.-> UC7 : <<include>>

        UC5 -.-> UC6 : <<include>>
        UC6 -.-> UC8 : <<include>>
        UC8 -.-> UC9 : <<include>>
        UC9 -.-> UC11 : <<include>>
        UC10 -.-> UC11 : <<include>>
        UC11 -.-> UC13 : <<include>>
        UC13 -.-> UC15 : <<include>>
        UC15 -.-> UC16 : <<include>>
        UC16 -.-> UC8 : <<include>>
        UC16 -.-> UC9 : <<include>>
        UC16 -.-> UC11 : <<include>>
        UC16 -.-> UC13 : <<include>>
        UC16 -.-> UC15 : <<include>>
    
```

8

```
graph TD; A[Система розпізнавання облич] --- B[Підсистема додавання обличчя]; A --- C[Підсистема локалізації та ідентифікації обличчя]; A --- D[Підсистема роботи з базою даних]; A --- E[Підсистема виведення на екран]; D --- F[(Базу даних)];
```

The diagram illustrates the architecture of a face recognition system. At the top is the main system box, 'Система розпізнавання облич' (Face Recognition System). It is connected to four sub-systems: 'Підсистема додавання обличчя' (Face Addition Subsystem), 'Підсистема локалізації та ідентифікації обличчя' (Face Localization and Identification Subsystem), 'Підсистема роботи з базою даних' (Database Management Subsystem), and 'Підсистема виведення на екран' (Screen Output Subsystem). The 'Підсистема роботи з базою даних' is further connected to a database icon, represented by a stack of cylinders.

Рисунок В.8 – Слайд 8

Діаграма класів програмного забезпечення

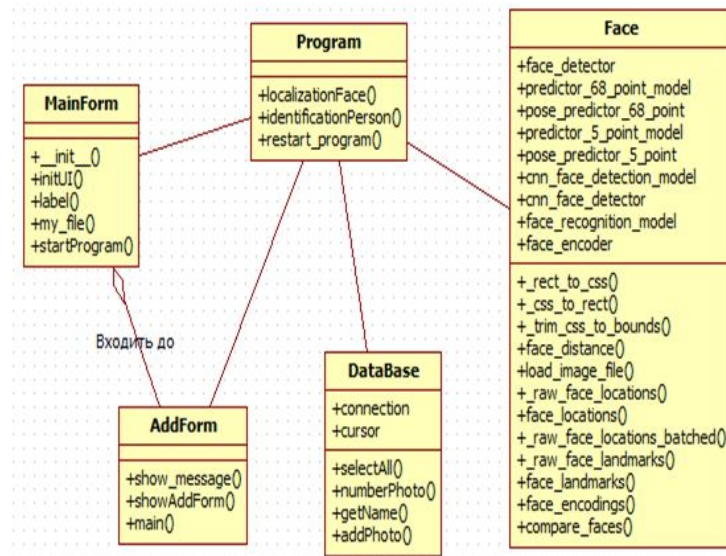


Рисунок В.9 – Слайд 9

Діаграма компонентів програмного забезпечення

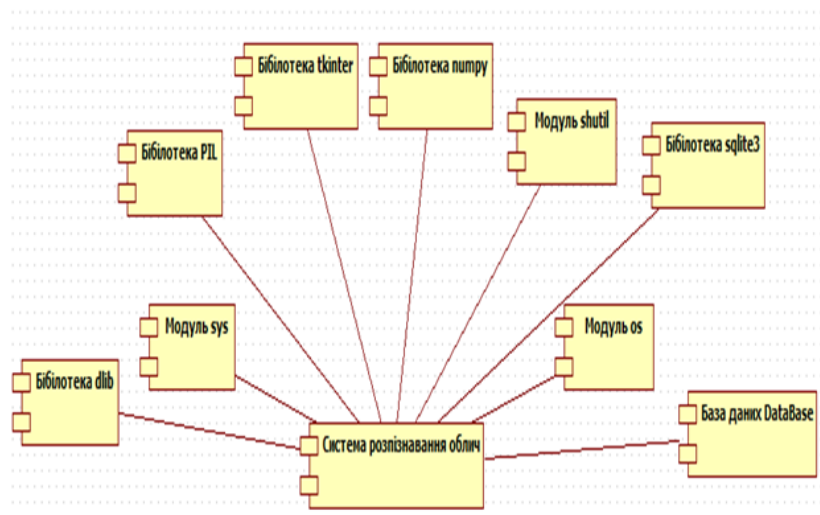


Рисунок В.10 – Слайд 10

Робота з програмою. Головне вікно програми з обраним фото для розпізнавання



Рисунок В.11 – Слайд 11

Робота з програмою. Діалогове вікна вибору зображення для локалізації обличчя. Результати локалізації

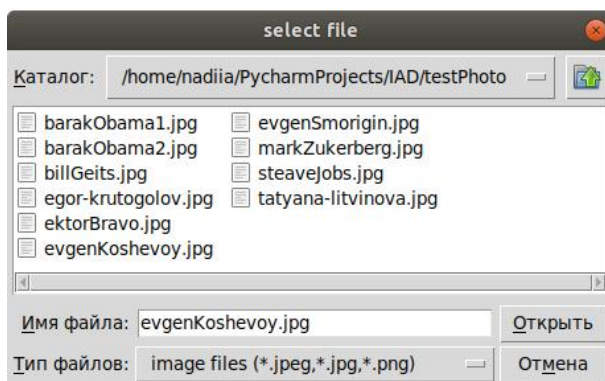


Рисунок В.12 – Слайд 12

Робота з програмою. Форма додавання фотографії нової особи до бази даних програми



Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для обробки зображень.

Новизна роботи полягає в тому, що запропоновано модель взаємодії користувача з програмним забезпеченням для локалізації облич на зображеннях, яка подана у вигляді діаграми прецедентів у нотації UML, та дозволяє розроблювати програмне забезпечення для локалізації облич на зображеннях.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для локалізації облич на зображеннях.

Результати тестування програмного забезпечення для локалізації облич на зображеннях показали, що розроблена програма дозволяє розв'язувати завдання локалізації та розпізнавання об'єктів на зображеннях.

Рисунок В.14 – Слайд 14