

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентці Герасимовій Світлані Станіславівні
(прізвище, ім'я, по батькові)1. Тема роботи Розробка застосунку для онлайн розпізнавання облич у відеопотоці

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 28 травня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, методи розпізнавання облич, HOG, SVM, Face Landmark Estimation, інтегроване середовище розробки PyCharm, бібліотека комп'ютерного зору з відкритим кодом OpenCV, пакет комп'ютерного зору CVZone, платформа Firebase.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Теоретичні основи розпізнавання облич _____

2. Аналіз методу HOG у задачі визначення облич _____

3. Моделювання методу для визначення ключових точок облич за допомогою алгоритму Face Landmark Estimation _____

4. Математична основа простого лінійного класифікатора SVM у задачі класифікації образів _____

5. Комп'ютерна модель системи розпізнавання облич _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розпізнавання облич, постановка задачі, блок-схема алгоритму розпізнавання облич, блок-схема етапів розробки застосунку, шаблони застосунку, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	10.04.24-15.04.24	
3	Аналіз літератури з досліджуваної проблеми	16.04.24-20.04.24	
4	Аналіз технічних засобів	20.04.24-25.04.24	
5	Вибір методів	26.04.24-30.05.24	
6	Програмна реалізація	01.05.24-23.05.24	
7	Оформлення пояснювальної записки	20.05.24-28.05.24	
8	Перевірка на плагіат	29.05.24	
9	Рецензування	30.05.24	
10	Підготовка презентації та доповіді	01.06.24-05.06.24	
11	Занесення роботи в електронний архів	06.06.24	
12	Попередній захист кваліфікаційної роботи	06.06.24	

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Шафроненко А.Ю.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 64 с., 2 табл., 56 рис., 32 джерела.

РОЗПІЗНАВАННЯ ОБЛИЧ, ДЕТЕКТУВАННЯ ОБЛИЧ, МАШИННЕ НАВЧАННЯ, ВЕБКАМЕРА, HOG, SVM, FACE LANDMARK ESTIMATION, КОМП'ЮТЕРНИЙ ЗІР, OPENCV, CVZONE.

Об'єктом роботи є проблема розпізнавання облич у реальному часі.

Метою роботи є розробка високоефективного та швидкодіючого застосунку, який здатний реалізовувати розпізнавання облич у реальному часі на відеопотоці.

Використано методи виявлення та розпізнавання обличчя, а також алгоритми машинного навчання. Проаналізовано методи виявлення облич на основі Histogram of Oriented Gradients та точну ідентифікацію ключових точок облич за допомогою алгоритму Face Landmark Estimation. Розглянуто математичну основу простого лінійного класифікатора SVM для класифікації образів облич. Проведено дослідження ефективності алгоритмів та їхню здатність виявляти обличчя, визначати ключові точки та класифікувати обличчя у режимі реального часу.

У результаті роботи здійснена програмна реалізація системи для онлайн розпізнавання облич у відеопотоці з використанням мови програмування Python та використанням бібліотек OpenCV і CVZone.

FACE RECOGNITION, FACE DETECTION, MACHINE LEARNING, WEBCAM, HOG, SVM, FACE LANDMARK ESTIMATION, COMPUTER VISION, OPENCV, CVZONE.

The object of work is the problem of real-time face recognition.

The aim of the work is to develop a highly efficient and fast application capable of real-time face recognition on a video stream.

The methods used are face detection and recognition, as well as machine learning algorithms. The methods of face detection based on Histogram of Oriented Gradients and accurate identification of key points of faces using the Face Landmark Estimation algorithm are analyzed. The mathematical basis of a simple linear SVM classifier for face image classification is considered. The effectiveness of the algorithms and their ability to detect faces, identify key points and classify faces in real time are investigated.

As a result of the work, a software implementation of the system for online face recognition in a video stream was implemented using the Python programming language and the OpenCV and CVZone libraries.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Теоретичні основи розпізнавання облич	8
1.1 Сучасний стан у сфері детектування: досягнення, проблеми та застосування	8
1.2 Основні етапи процесу виявлення осіб	9
1.3 Огляд існуючих методів виявлення облич.....	12
1.3.1 Методи на основі машинного навчання	12
1.3.2 Методи на основі відслідковування облич.....	13
1.3.3 Методи на основі геометричних характеристик.....	14
1.3.4 Методи на основі глибинного аналізу	15
1.3.5 Методи на основі відзначення ключових точок	16
1.4 Формування вимог до системи розпізнавання облич	17
1.5 Постановка задачі	18
2 Математичні моделі розпізнавання облич.....	20
2.1 Аналіз методу HOG у задачі визначення облич	20
2.2 Моделювання методу для визначення ключових точок облич за допомогою алгоритму Face Landmark Estimation	26
2.3 Математична основа простого лінійного класифікатора SVM у задачі класифікації образів.....	30
3 Комп'ютерна модель системи Розпізнавання облич.....	36
3.1 Обґрунтування вибору середовища програмної реалізації	36
3.2 Програмна реалізація системи розпізнавання облич	38
3.3 Інструкція користувача	45
3.4 Тестування розробленої моделі.....	52
Висновки	62
Перелік джерел посилання	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

CNN – Convolutional Neural Network (згорткові нейронні мережі)

YOLO – You Look Only Once

SSD – Single Shot Multibox Detector (одноходовий багаторамковий виявляч)

R-CNN – Region-based Convolutional Neural Networks (обласні згорткові нейронні мережі)

HOG – Histogram of Oriented Gradients (гістограма напрямлених градієнтів)

SVM – Support Vector Machine (метод опорних векторів)

Face Landmark Estimation – оцінка ключових точок обличчя

БД – база даних

AR – Augmented Reality (доповнена реальність)

VR – Virtual Reality (віртуальна реальність)

FKDN – Facial Keypoint Detection Network (мережа виявлення ключових точок на обличчі)

One Millisecond Face Alignment with an Ensemble of Regression Trees – мілісекундне вирівнювання обличчя за допомогою ансамблю дерев регресії

RBF – Radial Basis Function (радіальна базисна функція)

OpenCV – Open Source Computer Vision Library

CVZone – бібліотека Python для роботи з комп'ютерним зором

WIDER FACE – розширений набір даних для детектування обличч

CelebA – CelebFaces Attributes Dataset

IDE – Integrated Development Environment (інтегроване середовище розробки)

ВСТУП

Розпізнавання облич є однією з найбільш швидкозростаючих і впливових технологій у галузі комп'ютерного зору та штучного інтелекту, що знаходить застосування у багатьох сферах, таких як забезпечення безпеки, ідентифікація особистості, керування доступом, аналіз поведінки споживачів, охорона правопорядку та багато інших. Вступаючи у світ, де цифрові технології набувають все більшого значення, можливість точно і швидко ідентифікувати особу на фото і відео контенті стає ключовим елементом сучасних систем безпеки та обслуговування клієнтів. Однак розпізнавання образів залишається складною задачею через мінливість умов освітлення, різноманітність ракурсів та міміки, а також наявність шумів у відеопотоці [1].

Актуальність роботи полягає у потребі розробки ефективних методів для онлайн аналізу відео в реальному часі з метою виявлення та ідентифікації осіб, що дозволяє не тільки підвищити безпеку, але й вдосконалити користувацький досвід, забезпечуючи більш персоналізовані та інтерактивні послуги.

Робота присвячена розробці застосунку, який використовує передові методи обробки зображень для визначення осіб у різноманітних умовах освітлення та при різних ракурсах камери. Це включає використання алгоритмів, здатних визначати обличчя на зображенні, точно визначати розташування ключових елементів обличчя та класифікувати ці обличчя за допомогою методів машинного навчання, що забезпечують високу точність і швидкість обробки даних [2].

У роботі розглядається комплексний підхід, що включає обробку відеопотоку та детальний аналіз визначених осіб з метою ідентифікації та класифікації. Реалізація такого підходу дозволяє не тільки покращити наявні системи, але й відкриває нові можливості для розробки інноваційних застосунків у широкому спектрі областей застосування, забезпечуючи більшу безпеку, контроль та зручність користування.

1 ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ ОБЛИЧ

1.1 Сучасний стан у сфері детектування: досягнення, проблеми та застосування

За останні роки технологія розпізнавання облич досягла значного прогресу, трансформувавши низку сфер і ставши необхідним елементом повсякденного життя. Розпізнавання осіб має широкий спектр застосувань, що постійно розвиваються, від допомоги в медичній діагностиці до вдосконалення систем безпеки і розблокування мобільних телефонів.

Досягнення штучного інтелекту, особливо в галузі глибокого навчання, значно підвищили точність і ефективність систем розпізнавання облич. Зокрема, згорткові нейронні мережі (CNN) стали основою найсучасніших систем детектування. Такі моделі, як YOLO, SSD та R-CNN, продемонстрували чудову ефективність у точному та швидкому розпізнаванні осіб у різноманітних станах, включаючи фото та відео [3].

Крім того, навчання надійних моделей розпізнавання облич полегшилося завдяки наявності великих анотованих наборів даних, таких як WIDER FACE і CelebA. Використовуючи ці набори даних, вчені та програмісти можуть тренувати алгоритми на різноманітних рисах обличчя, виразах і сценаріях освітлення, покращуючи загальність і стійкість систем ідентифікації осіб [4, 5].

Розпізнавання облич все ще стикається з низкою проблем, незважаючи на значний прогрес, особливо в практичному застосуванні. За складних зовнішніх обставин традиційні підходи часто не можуть надійно розпізнавати обличчя, що призводить до помилкових спрацьовувань або пропущених виявлень. Забезпечення конфіденційності та дотримання етичних міркувань при впровадженні алгоритмів розпізнавання облич є ще однією складністю. Крім того, упередженість алгоритмів і навчальних даних може призвести до

несправедливих або дискримінаційних результатів, особливо коли йдеться про вік, стать і расу.

Незважаючи на існуючі перешкоди, технологія розпізнавання облич продовжує розвиватися завдяки передовим системам і новим тенденціям. Поєднання розпізнавання облич з іншими технологіями штучного інтелекту, дозволяє створювати все більш складні програми, які допомагають у вирішенні багатьох повсякденних питань.

Поточний стан розпізнавання облич відображає динамічний світ, що характеризується швидким розвитком, постійними проблемами та різноманітними застосуваннями. Оскільки технологія продовжує розвиватися, вирішення проблем, пов'язаних з точністю, конфіденційністю та упередженістю, буде мати важливе значення для зміцнення довіри та прийняття рішень щодо розпізнавання облич у суспільстві. Завдяки використанню потенціалу штучного інтелекту та етичних принципів розпізнавання облич має величезні перспективи для трансформації індустрії, підвищення безпеки та покращення добробуту людей.

1.2 Основні етапи процесу виявлення осіб

Існують різні підходи до розпізнавання облич, але незалежно від обраного методу, процес цього алгоритму включає кілька основних етапів. Розуміння цих етапів дозволяє краще усвідомити суть та складність завдання розпізнавання облич і забезпечити ефективну реалізацію алгоритмів у практичних застосуваннях.

Перший етап передбачає виявлення та визначення місцезнаходження обличчя особи на поточному зображенні або відео. Для зручності та точності подальших маніпуляцій, обличчя вирівнюється та стандартизується. Далі обчислюються ознаки, і отримані ознаки порівнюються з еталонними, які містяться в базі даних зображень облич [6].

Загальний алгоритм для розпізнавання обличчя представлено на рисунку 1.1.



Рисунок 1.1 – Чотири основні етапи розпізнавання обличчя

Розглянемо кожен крок більш детально на прикладі розпізнавання обличчя комедійного американського актора Вілла Феррелла (Will Ferrell):

Крок 1. Розпізнавання обличчя. Детектор обличчя визначає координати обмежувальної рамки для кожного обличчя після того, як їх місцезнаходження виявлено на зображенні (рис. 1.2).



Рисунок 1.2 – Рамка визначення обличчя [7]

Крок 2. Попередня обробка обличчя. Використовуються опорні точки та лінії на виявлених обличчях для рівномірного масштабування, вирівнювання та обрізання зображення, щоб забезпечити стандартний вигляд усіх зображень облич (рис. 1.3).

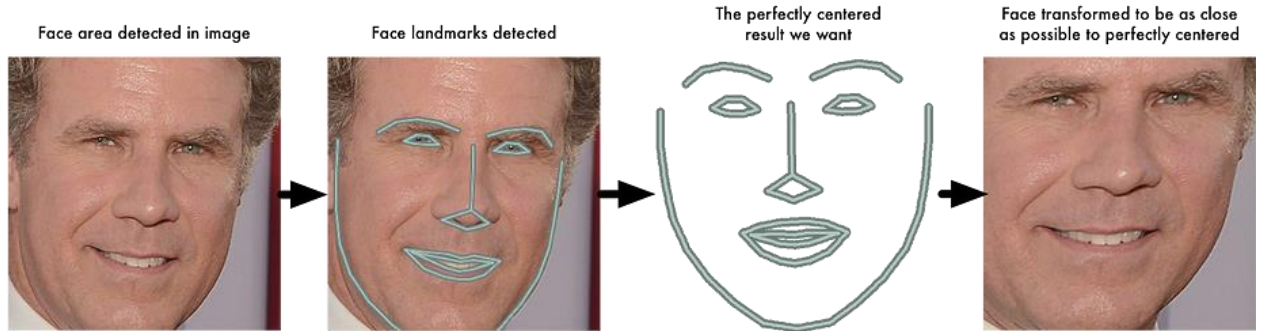


Рисунок 1.3 – Етапи вирівнювання обличчя [7]

Крок 3. Представлення обличчя. На цьому етапі значення пікселів зображення обличчя перетворюються у векторний елемент, відомий як шаблон, що представляє обличчя у форматі коду (рис. 1.4). В ідеалі всі грані одного і того ж об’єкта повинні відображатись у однакових векторах об’єктів [6].

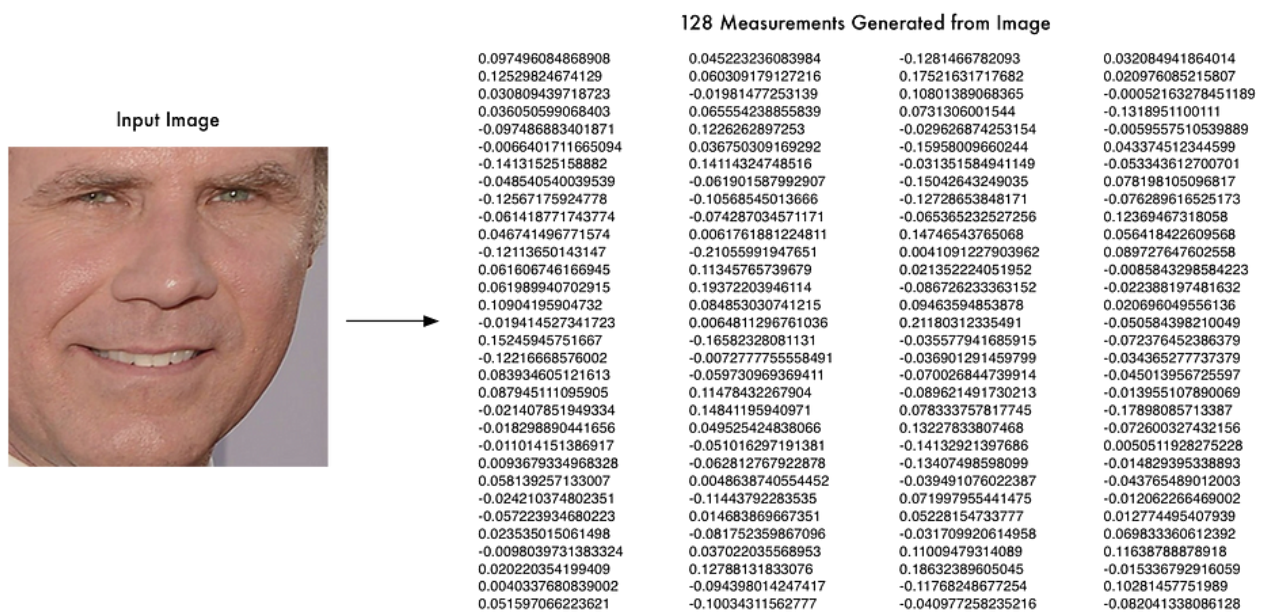


Рисунок 1.4 – Вектор зі 128 елементів згенерованих з фото обличчя [7]

Крок 4. Порівняння ознак облич. Два шаблони порівнюються для визначення ступеня їхньої схожості. Цей процес допомагає встановити ймовірність того, що різні шаблони належать одній і тій же особі (рис. 1.5).

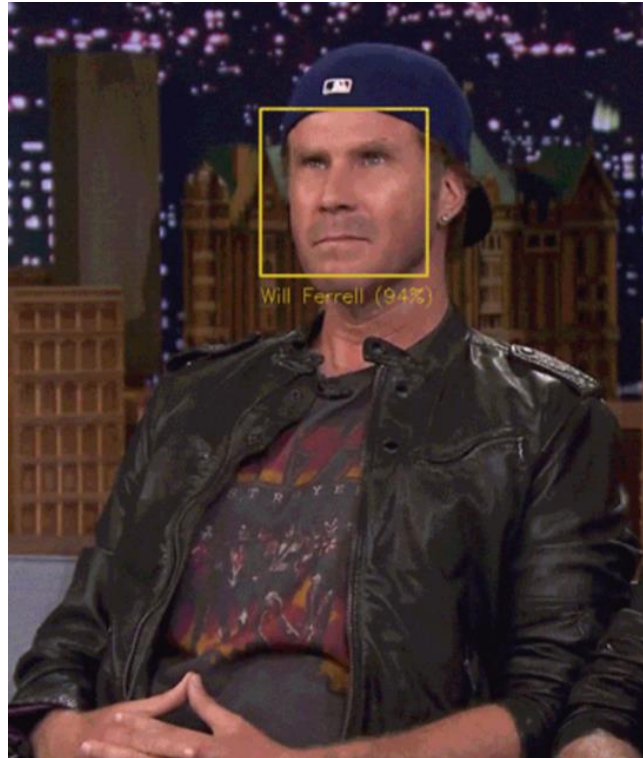


Рисунок 1.5 – Результат та точність схожості шаблонів облич [7]

1.3 Огляд існуючих методів виявлення облич

1.3.1 Методи на основі машинного навчання

Методи, що базуються на машинному навчанні, використовують алгоритми глибокого навчання, зокрема згорткові нейронні мережі (CNN), для виявлення облич у зображеннях. CNN – це моделі глибокого навчання, спроектовані на основі роботи зорової кори людини, які здатні вивчати складні патерни та особливості на основі необроблених піксельних даних. Ці мережі складаються з декількох шарів, включаючи згорткові шари, об'єднуючі шари та повністю пов'язані шари, які працюють синергетично для вилучення ієрархічних представлень вхідних даних [8–11].

Спочатку модель навчається на великих масивах даних, що містять зображення облич з різних ракурсів, поз та умов освітлення. Під час навчання мережа вчиться ідентифікувати ключові риси обличчя, такі як очі, ніс, рот, а також їхнє просторове розташування. Як результат, навчений CNN стає досвідченим у розрізненні облич та об'єктів на фото та відео контенті [9].

Методи розпізнавання облич на основі машинного навчання мають численні практичні застосування в різних сферах. У сфері безпеки та відеоспостереження вони використовуються для верифікації особи, контролю доступу та моніторингу громадських місць на предмет загроз безпеці. У фотографії та соціальних мережах розпізнавання облич уможливорює автоматичне тегування, розпізнавання облич для впорядкування фотоколекцій та застосування фільтрів доповненої реальності.

Перевагою цього методу є висока точність розпізнавання, особливо при навчанні на великому і різноманітному наборі даних. Однак навчання моделі потребує значних обчислювальних ресурсів і великої кількості навчальних даних.

1.3.2 Методи на основі відслідковування облич

Ці методи використовують алгоритми для виявлення та прогнозування траєкторій руху облич на послідовних кадрах відео або послідовності зображень. Вони можуть застосовувати численні стратегії, доповнені алгоритмами машинного навчання, що надає надійні та точні рішення для відстеження облич, незважаючи на такі проблеми, як варіації освітлення, пози і оклюзії. Мета полягає в тому, щоб визначити рух обличчя в кожному кадрі відео та відстежити його траєкторію [12].

Застосування методів на основі відслідковування облич різноманітне і широко розповсюджене. У застосунках доповненої реальності (AR) і віртуальної реальності (VR) відстеження обличчя дозволяє накладати

цифровий контент на обличчя користувача в режимі реального часу, створюючи ефект занурення. У системах взаємодії між людиною та комп'ютером, таких як відстеження погляду або розпізнавання емоцій, відслідковування облич забезпечує цінний внесок у розуміння поведінки та намірів користувача. Крім того, в системах спостереження і безпеки відстеження облич полегшує безперервний моніторинг і аналіз переміщень окремих осіб в умовах великого скупчення людей.

Однією з переваг підходу відстеження є можливість оцінити рух і поведінку людей на відео. Однак цей метод може бути складним у застосуванні і вимагати значних обчислювальних ресурсів, особливо в сценаріях з великою кількістю об'єктів або відео з низькою роздільною здатністю.

1.3.3 Методи на основі геометричних характеристик

Геометричні методи ґрунтуються на виділенні та аналізі особливості облич, такі як розмір, форма, текстура та відстань між ключовими точками [13].

Ключовою концепцією розпізнавання облич за геометричними ознаками є використання шаблонів або моделей, які описують геометричну структуру обличчя. Ці шаблони часто складаються із заздалегідь визначених рис обличчя, таких як очі, ніс, рот і загальний контур. Порівнюючи розташування цих елементів на зображенні з очікуваним шаблоном, методи, засновані на геометричних ознаках, можуть виявляти і локалізувати обличчя.

Одним із поширених підходів до розпізнавання облич за геометричними ознаками є використання алгоритму Віоли-Джонса, в якому застосовуються ознаки типу Хаара та каскадні класифікатори. Хаар-об'єкти – це прості прямокутні шаблони, які фіксують варіації інтенсивності пікселів у межах області, що цікавить. Алгоритм сканує зображення за допомогою каскаду

класифікаторів, кожен з яких фокусується на різних особливостях і поступово усуває нелицьові області на основі набору заздалегідь визначених критеріїв [14–16].

Перевагами таких методів є простота реалізації та менша залежність від великих навчальних наборів, що робить їх придатним для застосування в реальному часі на пристроях з обмеженими ресурсами. Однак методи на основі геометричних характеристик можуть бути менш точними у порівнянні з методами на основі машинного навчання. Ці алгоритми значною мірою покладаються на точність заздалегідь визначених шаблонів облич, які можуть не дуже добре узагальнювати різноманітні та складні сценарії реального світу.

1.3.4 Методи на основі глибинного аналізу

Методи на основі глибинного аналізу використовують інформацію про глибину зображення, отриману зі стереокамер або сенсорів глибини, для виявлення облич. На відміну від традиційних методів, заснованих на двовимірних зображеннях, ці підходи використовують додаткову інформацію про тривимірну структуру об'єктів у сцені, що дозволяє точніше розпізнати обличчя.

Основою роботи таких методів є вимірювання глибини, яке передбачає збір інформації про глибину разом з RGB-даними за допомогою спеціалізованих датчиків, таких як камери Time-of-Flight або датчики структурованого світла. Ці датчики випромінюють інфрачервоне світло на сцену і вимірюють час, за який світло повертається, що дозволяє обчислити глибину кожного пікселя на зображенні [17, 18].

Для розпізнавання облич з використанням інформації про глибину було запропоновано кілька підходів, включаючи традиційні методи, такі як вилучення ознак на основі глибини в поєднанні з класифікаторами машинного навчання, а також методи на основі глибокого навчання, пристосовані для

використання глибинних підказок. Архітектури глибокого навчання, такі як тривимірні згорткові нейронні мережі (CNN), можуть безпосередньо обробляти дані про глибину, щоб вивчати дискримінаційні ознаки для розпізнавання облич, що призводить до покращення продуктивності порівняно з двовимірними CNN [10].

Крім того, методи глибинного аналізу забезпечують підвищену стійкість до атак підміни та спроб видавання себе за іншого, оскільки датчики глибини можуть фіксувати додаткову геометричну інформацію, яку складно відтворити, використовуючи лише традиційні 2D-зображення. Це робить системи розпізнавання облич на основі глибинного аналізу більш надійними для застосунків, що вимагають високого рівня безпеки та автентифікації, таких як біометричний контроль доступу та платіжна автентифікація.

Незважаючи на свої численні переваги, методи глибинного аналізу також створюють певні проблеми та обмеження. Датчики глибини відносно дорогі і не завжди доступні на всіх пристроях, що обмежує широке впровадження цих технологій.

1.3.5 Методи на основі відзначення ключових точок

Методи, що використовують ключові точки, фокусуються на критичних місцях обличчя, такі як куточки очей, носа, рота та брів, для розпізнавання та відстеження облич. Вони можуть використовувати алгоритми локального виявлення ознак або глобального зіставлення зображень, щоб встановити положення цих ключових точок у кожному кадрі відео.

Однією з популярних архітектур для виявлення ключових точок на обличчі є мережа виявлення ключових точок на обличчі FKDN, яка складається з декількох згорткових і об'єднуючих шарів, за якими слідує повністю з'єднані шари для задач регресії або класифікації. FKDN навчається на великомасштабних наборах даних, що містять анотовані ключові точки

обличчя, що дозволяє мережі вивчати просторові взаємозв'язки та відмінні риси ключових орієнтирів обличчя [19].

Іншим перспективним напрямком у виявленні ключових точок для розпізнавання облич є інтеграція механізмів уваги та мереж просторових трансформаторів. Механізми уваги дозволяють нейронним мережам фокусуватися на релевантних областях вхідного зображення, що сприяє більш точній і надійній локалізації ключових точок. Мережі просторових трансформаторів уможливають геометричні перетворення карт ознак, що дозволяє мережі адаптивно вирівнювати та виправляти ділянки обличчя для покращення виявлення ключових точок [20].

Таких метод є ефективним, коли обличчя частково закриті або змінюють свою позу. Однак він може бути чутливим до змін освітлення, виразів обличчя та інших факторів.

1.4 Формування вимог до системи розпізнавання облич

Система розпізнавання облич повинна бути насамперед точною та швидкою. При розпізнаванні облич з БД осіб система повинна заносити дані у реальному часі у БД системи відвідування з усією інформацією про розпізнану особу. Система має передбачати, що одна особа може зареєструвати своє відвідування не частіше, ніж один раз у добу. Інтерфейс програмного застосунку має бути привабливим та інтуїтивним у використанні.

Функціональні вимоги до системи включають:

- обробка вхідного відеопотоку повинна відбуватися у реальному часі без помітної затримки;
- висока точність розпізнавання облич у різних умовах освітлення та фонових умовах;
- точне розпізнавання облич при зміні поз та при наявності додаткових аксесуарів на особі;

- можливість інтеграції з іншими системами та базами даних;
- налаштування рівнів доступу для користувачів системи з різними правами;
- постійний доступ до бази даних з фото осіб для порівняння та ідентифікації;
- ідентифіковане обличчя необхідно пов'язувати з відповідними записами у БД.

Нефункціональні вимоги до системи є наступними:

- висока точність у розпізнаванні та ідентифікації облич;
- швидка та ефективна робота системи навіть при великому обсязі даних та високому навантаженні;
- масштабованість для роботи з різними обсягами даних та потоками відео;
- стабільність роботи та відсутність збоїв;
- зручне та швидке оновлення системи без порушення її роботи;
- захист конфіденційності та цілісності даних користувачів у БД;
- доступність для користувачів у будь-який час та з будь-якого пристрою з підключенням до мережі.

1.5 Постановка задачі

Таким чином, розпізнавання облич є актуальним завданням для багатьох сфер життя. Тому ставиться завдання дослідити та вдосконалити можливості ефективної онлайн-системи розпізнавання облич, що працює у відеопотоці.

Об'єктом роботи є проблема розпізнавання облич у реальному часі.

Метою роботи є розробка високоефективного та швидкодіючого застосунку, який здатний реалізовувати розпізнавання облич у реальному часі на відеопотоці.

Для виконання поставленої задачі було обрано методи HOG для виявлення обличчя, Face Landmark Estimation для точної ідентифікації ключових точок обличчя та простий лінійний класифікатор SVM для класифікації обличчя.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів розпізнавання облич;
- дослідити та вибрати оптимальні параметри для методів HOG та Face Landmark Estimation для максимально точного та ефективного розпізнавання облич у відеопотоці;
- реалізувати алгоритми HOG та Face Landmark Estimation для видобуття ознак облич у відеопотоці за допомогою бібліотеки OpenCV та CVZone;
- підготувати навчальний датасет для навчання SVM класифікатора, що включатиме зображення з різних ракурсів та умов освітлення;
- навчити простий лінійний SVM класифікатор за допомогою навчального датасету, щоб він міг ідентифікувати обличчя на відеопотоці;
- інтегрувати реалізовані алгоритми у вебзастосунок, який буде здатний обробляти відеопотік у реальному часі та відображати результати розпізнавання облич на відеопотоці;
- розробити графічний інтерфейс застосунку для зручного користування системою;
- розробити БД для збереження даних розпізнаних осіб;
- здійснити тестування розробленого застосунку на різних відео з різними умовами освітлення та фоном для оцінки точності та швидкодії розпізнавання облич;
- провести порівняльний аналіз розробленого застосунку з існуючими аналогами за критеріями точності, швидкодії та стійкості до змін умов;
- задокументувати процес розробки, включаючи опис алгоритмів, використані бібліотеки, налаштування параметрів та результати тестування.

2 МАТЕМАТИЧНІ МОДЕЛІ РОЗПІЗНАВАННЯ ОБЛИЧ

2.1 Аналіз методу HOG у задачі визначення облич

Метод Histogram of Oriented Gradients є однією з популярних технік для розпізнавання облич в зображеннях. Розроблений вперше Далалом і Тріггс у 2005 році, цей метод широко використовується через його ефективність у виявленні форм і об'єктів. HOG описує місцеві об'єкти зображення через розподіл напрямків або орієнтацій градієнтів [21].

Основна ідея методу HOG полягає в тому, що зовнішній вигляд і форма місцевого об'єкта в зображенні може бути описана достатньо добре за допомогою розподілу інтенсивності градієнтів або країв. Це досягається шляхом поділу зображення на невеликі з'єднані регіони, звані клітинами, і обчислення гістограми градієнтних напрямків або розподілу країв для кожної клітини.

Вектори градієнтів зображення визначаються для кожного пікселя, де величина та напрямок градієнта визначаються за формулами:

$$G_x = I(x + 1, y) - I(x - 1, y), \quad (2.1)$$

$$G_y = I(x, y + 1) - I(x, y - 1), \quad (2.2)$$

$$G = \sqrt{G_x^2 + G_y^2}, \quad (2.3)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right), \quad (2.4)$$

де G_x – горизонтальна компонента градієнта;

G_y – вертикальна компонента градієнта;

G – величина градієнта;

θ – напрямок градієнта.

Кожна клітина зображення складається з декількох пікселів, і для кожної клітини обчислюється гістограма градієнтів. Кількість сегментів у гістограмі може варіюватися, але типове значення становить 9. Гістограма орієнтованих градієнтів для клітини і регіональної блокується для кращої нормалізації [21].

Продовжимо практично розглядати HOG метод для виявлення облич на прикладі фото актора Вілла Феррелла (Will Ferrell).

Спочатку треба перетворити зображення на чорно-біле, оскільки для пошуку облич не потребуються кольорові дані (рис. 2.1).



Рисунок 2.1 – Перетворене зображення на чорно-біле [7]

Лістинг 2.1 Реалізація перетворення кольорового зображення на чорно-біле за допомогою бібліотеки OpenCV:

```
# Convert BGR (OpenCV default) to RGB  
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

Потім розглянемо кожен піксель зображення індивідуально. Для кожного пікселя необхідно подивитись на пікселі, які оточують його (рис. 2.2).

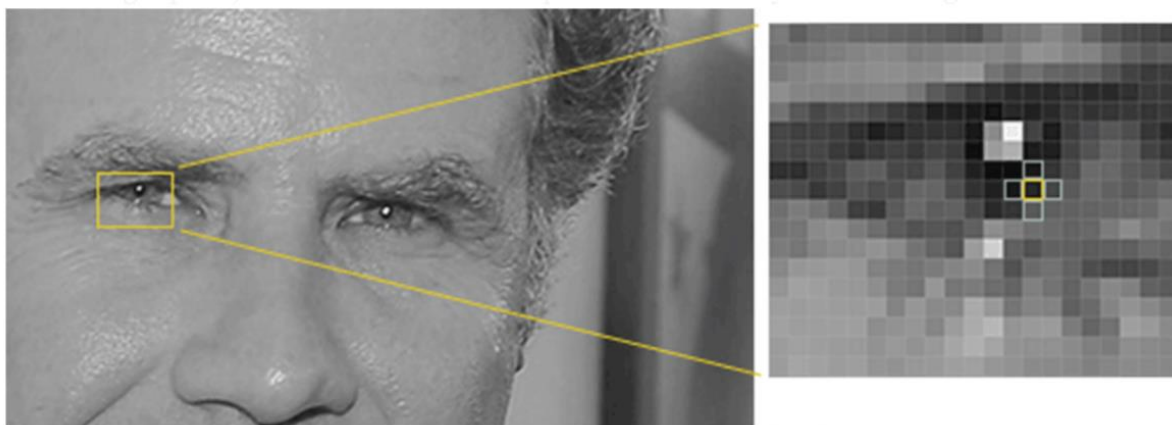


Рисунок 2.2 – Піксель на зображенні та оточуючі його пікселі [7]

Метою є з'ясувати, наскільки темним є поточний піксель у порівнянні з оточуючими його пікселями. Далі малюється стрілка, яка показує, в якому напрямку зображення стає темнішим (рис. 2.3).

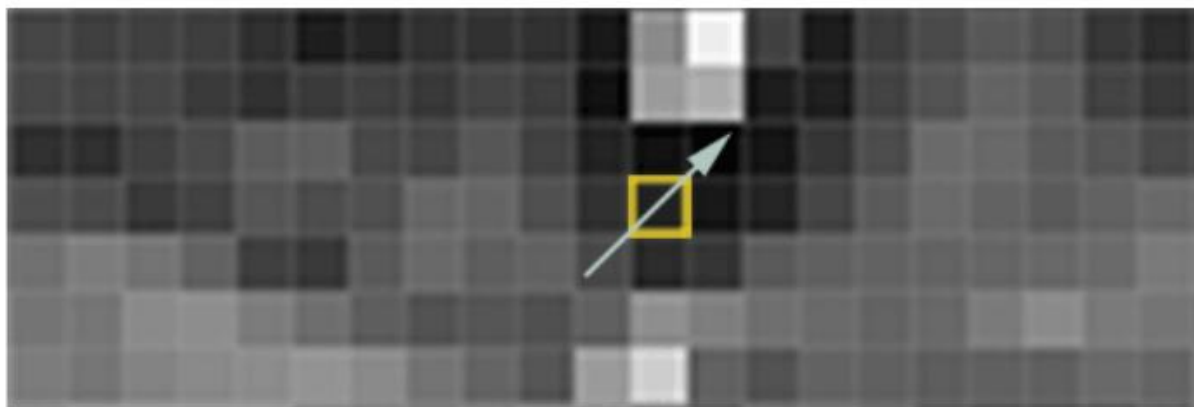


Рисунок 2.3 – Стрілка, яка вказує напрям затемнення від поточного пікселя [7]

Розглядаючи лише цей один піксель та пікселі, що оточують його, можна побачити, що зображення стає темнішим у напрямку правого верхнього кута. Якщо повторити цей процес для кожного пікселя, то усі пікселі зображення

буде замінено стрілкою. Ці стрілки називаються градієнтами і вони показують потік світла до темряви по всьому зображенню (рис. 2.4).

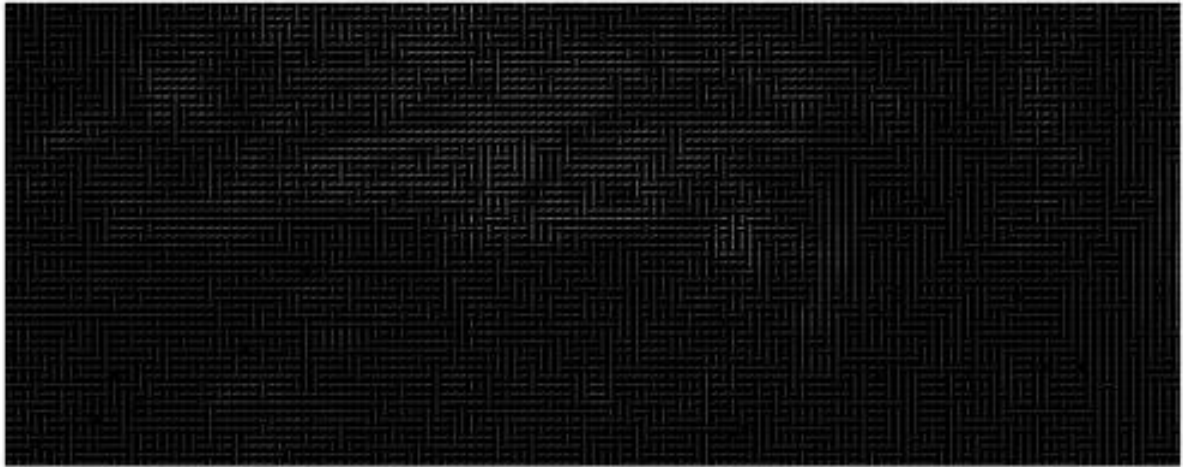


Рисунок 2.4 – Фрагмент градієнтного зображення [7]

Це процес може здатися малозначущим, але існує вагома причина для заміни пікселів градієнтами. Якщо аналізувати пікселі безпосередньо, дуже темні зображення та дуже світлі зображення однієї й тієї ж особи матимуть зовсім різні значення пікселів. Але, враховуючи лише напрямок зміни яскравості, як дуже темні, так і дуже яскраві зображення матимуть однакове представлення.

Зберігання градієнта для кожного окремого пікселя дає занадто багато деталей. Є ризик загубити загальний образ за окремими деталями. Краще бачити основний потік світла або темряви на вищому рівні, щоб була можливість розгледіти основний візерунок зображення.

Для цього зображення розділяється на маленькі квадрати по 16 на 16 пікселів кожен. У кожному квадраті рахується, скільки градієнтів вказують у кожному основному напрямку (скільки вказують вгору, вгору-вправо, вправо тощо). Потім цей квадрат замінюється у зображенні на напрямки стрілок, які були найсильнішими [22].

Кінцевий результатом є перетворення оригінального зображення на дуже просте представлення, яке демонструє основну структуру обличчя безпосереднім способом (рис. 2.5).

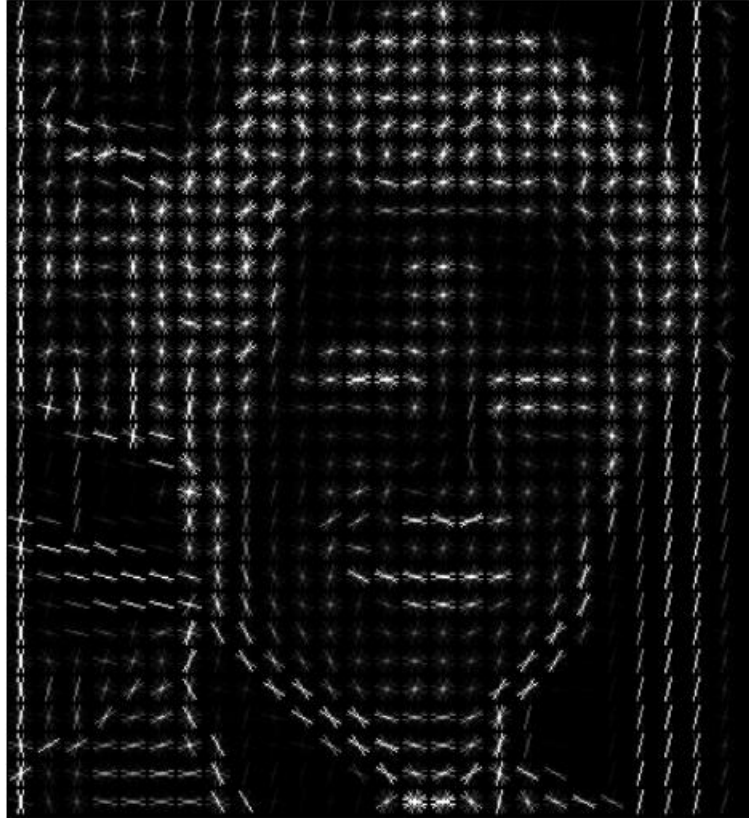


Рисунок 2.5 – Кінцевий вигляд HOG представлення [7]

Оригінальне зображення перетворюється на представлення HOG, яке розпізнає основні риси обличчя незалежно від яскравості зображення. Для виявлення обличчя у цьому зображенні HOG потрібно знайти частину зображення, яка найбільше схожа на відомий візерунок HOG, який був отриманий з БД інших тренувальних облич (рис. 2.6).

Лістинг 2.2 Приклад реалізації алгоритму розпізнавання обличчя HOG за допомогою вбудованого класу `dlib`:

```
# Create a HOG face detector using the built-in dlib class  
face_detector = dlib.get_frontal_face_detector()
```

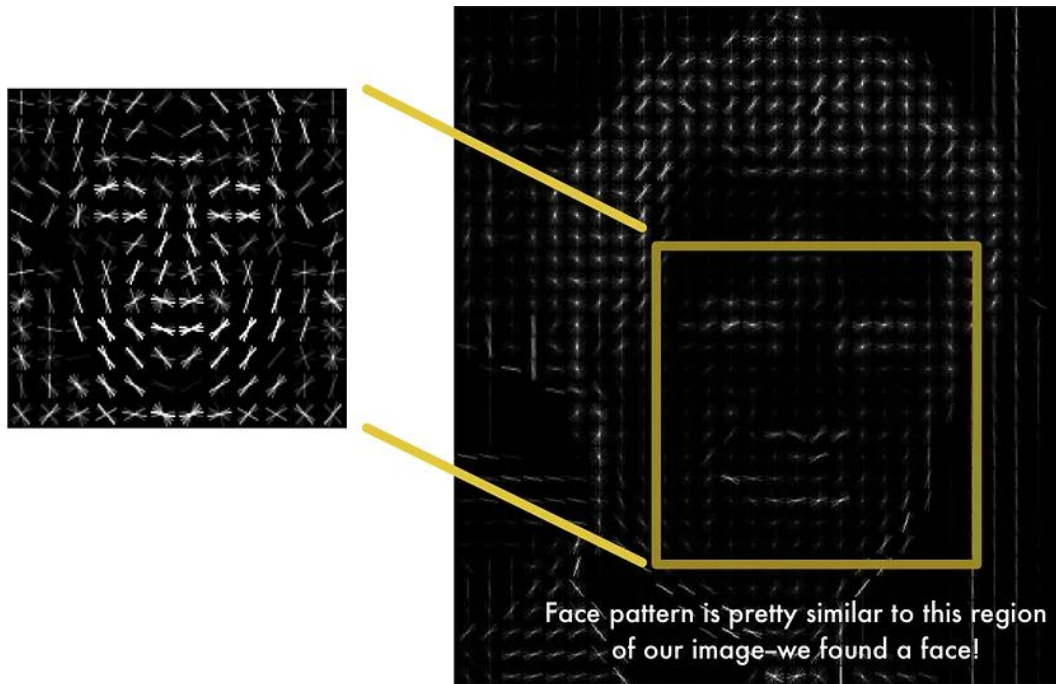


Рисунок 2.6 – Порівняння HOG зображень [7]

Результат знаходження обличчя на тестовому зображенні методом HOG показано на рисунку 2.7.

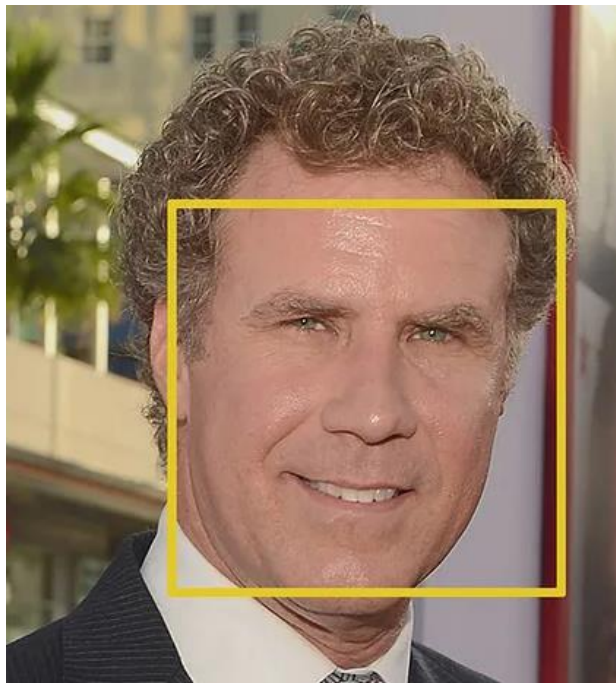


Рисунок 2.7 – Приклад знаходження обличчя на зображенні методом HOG [7]

2.2 Моделювання методу для визначення ключових точок облич за допомогою алгоритму Face Landmark Estimation

Визначення ключових точок облич є важливим аспектом у системах комп'ютерного зору і має широкий спектр застосувань, включаючи розпізнавання осіб, аналіз емоцій, анімацію та інші. Цей метод базується на виявленні в структурі облич специфічних орієнтирів, таких як куточки очей, рота, контури чола та інших.

Математичні моделі для оцінки ключових точок обличчя зазвичай базуються на методах машинного навчання і комп'ютерного зору. Одним з передових методів в цій галузі є алгоритм «One Millisecond Face Alignment with an Ensemble of Regression Trees» винайдений у 2014 році Вахідом Каземі та Жозефіною Салліван, який дозволяє швидко і точно виконувати оцінку ключових точок обличчя. Цей метод базується на використанні ансамблю регресійних дерев, який ефективно моделює співвідношення між інтенсивностями пікселів обличчя і положенням ключових точок [23].

Регресійні дерева є основою багатьох алгоритмів машинного навчання і використовуються для моделювання залежностей між даними через навчання дерева рішень, де листя відповідають прогнозним значенням, а вузли – точкам рішення на основі вхідних характеристик.

Задача оцінки ключових точок може бути сформульована як задача регресії:

$$p^x = \arg \min_p \sum_{i=1}^n \| f(x_i; p) - y_x \|^2, \quad (2.5)$$

де p – параметри моделі;

x_i – вхідні дані (інтенсивності пікселів);

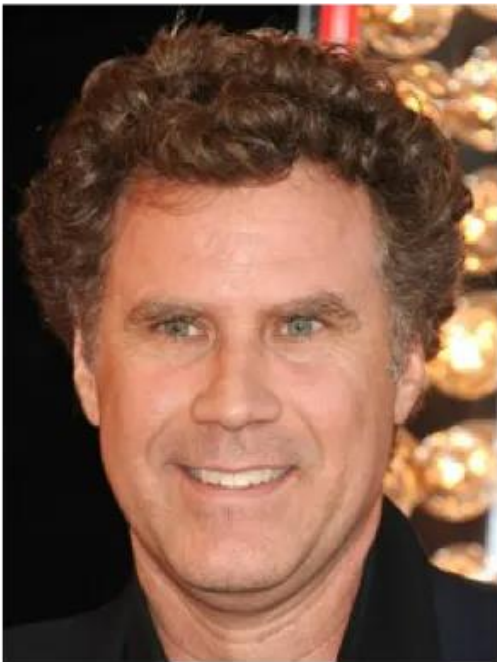
y_i – вектори цільових ключових точок;

$f(x_i; p)$ – функція моделі, що прогнозує положення ключових точок [24].

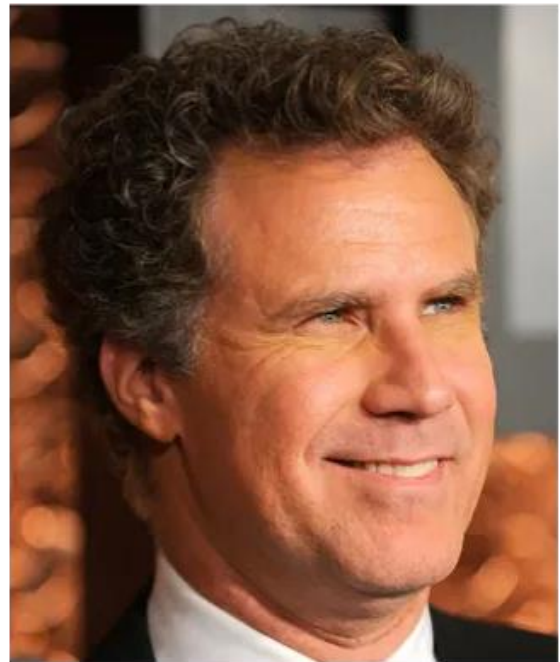
Модель включає набір регресійних дерев, де кожне дерево навчається на основі вибірки з навчального датасету. Використовується метод бутстрепінгу для генерації різних підвбірок даних для кожного дерева. Такий підхід дозволяє зменшити варіативність і покращити стабільність прогнозування.

Розглянемо практичне використання алгоритму визначення ключових точок на обличчі використовуючи фото актора Вілла Феррелла (Will Ferrell).

Люди легко розпізнають, що на обох рисунках 2.8 (а) та 2.8 (б) Вілл Феррелл, але комп'ютери бачитимуть на цих фото дві абсолютно різні особи. Щоб врахувати такі аспекти, необхідно трансформувати кожне зображення так, щоб очі та губи завжди були на одному місці на зображенні.



(а)



(б)

Рисунок 2.8 – Приклади фото Вілла Феррелла з різних ракурсів [7]:

(а) фото у профіль; (б) фото в анфас

Основна ідея полягає в тому, що треба визначити 68 конкретних точок (landmarks), які існують на кожному обличчі – верхня частина підборіддя, зовнішній край кожного ока, внутрішній край кожної брови тощо. Потім

навчити алгоритм машинного навчання виявляти ці 68 конкретних точок на будь-якому обличчі (рис. 2.9).

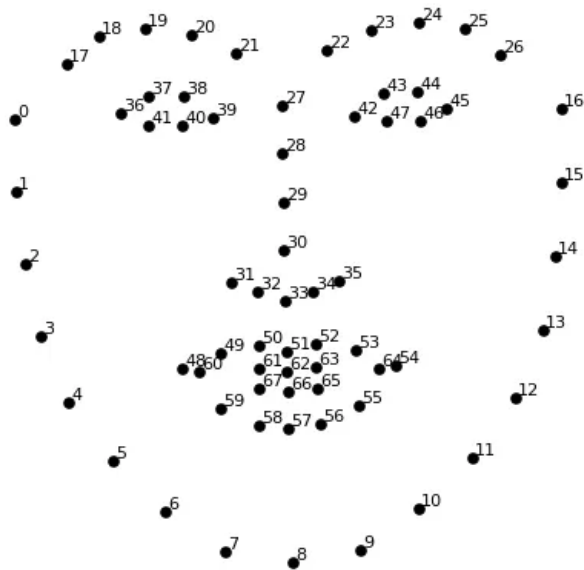


Рисунок 2.9 – 68 ключових точок обличчя [7]

Результат розташування 68 ключових точок обличчя на тестовому зображенні показано на рисунку 2.10.

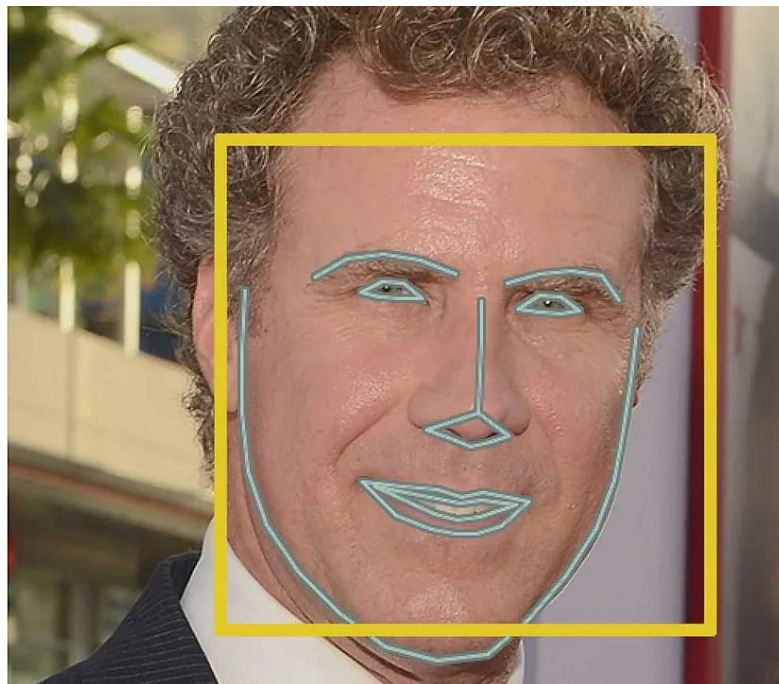


Рисунок 2.10 – Ключові точки на обличчі [7]

Лістинг 2.3 Приклад реалізації знаходження ключових точок обличчя за допомогою вбудованого класу `dlib`:

```
predictor_model = "shape_predictor_68_face_landmarks.dat"
face_pose_predictor = dlib.shape_predictor(predictor_model)
# Get the the face's pose
pose_landmarks = face_pose_predictor(image, face_rect)
```

Лістинг 2.4 Приклад реалізації знаходження ключових точок обличчя за допомогою бібліотеки `face_recognition`:

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```

Тепер, коли відомо, де знаходяться очі, ніс та рот на рисунку 2.11 (а), потрібно просто повернути, масштабувати та відкоригувати зображення так, щоб очі, ніс та рот були центровані найбільш точно. Немає потреби виконувати складні 3D трансформації, оскільки це може спотворити зображення. Натомість, будемо використовувати лише базові перетворення зображень, такі як обертання та масштабування, які зберігають паралельні лінії, як показано на рисунку 2.11 (б).

Лістинг 2.5 Приклад реалізації трансформації обличчя за допомогою вбудованого класу `dlib`:

```
predictor_model = "shape_predictor_68_face_landmarks.dat"
face_aligner = openface.AlignDlib(predictor_model)
# Use openface to calculate and perform the face alignment
alignedFace = face_aligner.align(534, image, face_rect, landmarkIndices =
openface.AlignDlib.OUTER_EYES_AND_NOSE)
```



(a)

(б)

Рисунок 2.11 – Перетворення обличчя за допомогою Face Landmark Estimation [7]:

(a) обличчя з виявленими орієнтирами; (б) трансформоване обличчя

Тепер незалежно від того, як повернене обличчя, є можливість центрувати очі, ніс та рот приблизно на одній і тій же позиції на зображеннях. Це значно спростить процес порівняння облич у наступних кроках.

2.3 Математична основа простого лінійного класифікатора SVM у задачі класифікації образів

Серед багатьох інструментів машинного навчання для класифікації та розпізнавання образів, лінійний класифікатор SVM (Support Vector Machine) вирізняється своєю ефективністю, особливо коли мова йде про розпізнавання облич.

SVM – це алгоритм машинного навчання, який можна використовувати для класифікації або регресії. Основна ідея полягає у визначенні гіперплощини в просторі ознак, яка найкраще розділяє два класи даних. У контексті

розпізнавання облич, ознаки можуть включати різні аспекти обличчя, які були перетворені в числовий формат за допомогою методів кодування облич [25].

Формула гіперплощини має наступний вигляд:

$$w^T x + b = 0, \quad (2.6)$$

де w – вектор вагових коефіцієнтів;

x – вектор ознак;

b – зсув.

Для того щоб знайти оптимальну гіперплощину, SVM вирішує задачу оптимізації, де мінімізується величина вектора w , забезпечуючи при цьому належне розділення між класами з деяким зазором. Це зазвичай формулюється як:

$$\min \frac{1}{2} \| w \|^2, \quad (2.7)$$

під умовами:

$$y_i (w^T x_i + b) \geq 1, \text{ для всіх } i, \quad (2.8)$$

де y_i – мітки класів, які приймають значення +1 або -1, залежно від класу до якого належить об'єкт;

x_i – вектори ознак для кожного об'єкта в наборі даних.

У випадку, коли дані не можуть бути лінійно розділені у вихідному просторі ознак, SVM може застосовувати так звані ядрові функції, щоб перевести дані в простір вищої розмірності, де розділення може бути лінійним. Ядрові функції дозволяють виконувати обчислення в просторі вищої розмірності без реального перетворення кожного вектора ознак, що значно спрощує вирахування та зберігає ресурси. Найпоширенішими ядровими

функціями є лінійне ядро, поліноміальне ядро, радіальна базисна функція (RBF) та сигмоїд. Кожна з цих функцій має свої переваги в залежності від структури даних та вимог до задачі класифікації, дозволяючи SVM адаптуватися до різних складностей у шаблонах даних [26–27].

Для розпізнавання облич часто використовують радіальну базисну функцію (RBF), яка визначається наступним чином:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad (2.9)$$

де γ – параметр, що визначає ширину ядра.

Опис ключових характеристик кожної ядрової функції показано у таблиці 2.1.

Таблиця 2.1 – Опис ядрових функцій

Ядрова функція	Опис	Приклади використання
Лінійне	Не додає додаткову розмірність; просте і ефективне для лінійно роздільних даних.	Прості набори даних, де розділення очевидне.
Поліноміальне	Може представляти взаємозв'язки між ознаками до заданого степеню.	Де потрібне зображення взаємозв'язків вищого порядку між ознаками.
Радіальна базисна функція (RBF)	Дуже ефективна для нелінійного розділення; може обробляти складні взаємодії між ознаками.	Комплексні набори даних, де лінійне розділення неможливе.
Сигмоїд	Перетворює вхідні дані в двійковий формат, подібно до нейронних мереж.	Використання в задачах, де потрібні властивості подібні до нейронних мереж.

Навчання SVM полягає у знаходженні вектора вагових коефіцієнтів та зсуву, які мінімізують вказану задачу оптимізації. Після тренування, класифікація нових прикладів виконується за допомогою оцінки знаку функції рішення.

$$f(x) = \text{sign}(w^T x + b). \quad (2.10)$$

Для розпізнавання облич за допомогою SVM, спочатку потрібно вибрати або створити набір ознак, які ефективно представляють унікальні характеристики кожного обличчя.

Далі, кожне обличчя у базі даних отримує мітку (наприклад, ім'я людини), і SVM використовується для навчання на цьому анотованому наборі даних. При отриманні нового обличчя система обробляє його для отримання відповідних ознак і застосовує тренований класифікатор для визначення, до якої мітки найбільш ймовірно належить дане обличчя (табл. 2.2).

Таблиця 2.2 – Результати класифікації SVM облич знаменитостей

№	Ідентифікатор образу	Справжнє ім'я	Передбачене ім'я	Відсоток схожості
1	2	3	4	5
1	001	Тейлор Свіфт	Тейлор Свіфт	98%
2	002	Бред Пітт	Бред Пітт	95%
3	003	Анджеліна Джолі	Скарлетт Йоханссон	85%
4	004	Леонардо Ді Капріо	Леонардо Ді Капріо	99%
5	005	Дженніфер Лоуренс	Дженніфер Лоуренс	82%
6	006	Джонні Депп	Джонні Депп	96%
7	007	Емма Стоун	Емма Стоун	97%
8	008	Раян Гослінг	Раян Рейнольдс	87%

Продовження таблиці 2.2

1	2	3	4	5
9	009	Селена Гомес	Селена Гомес	94%
10	010	Джордж Клуні	Джордж Клуні	93%
11	011	Кіра Найтлі	Кіра Найтлі	96%
12	012	Том Хенкс	Том Хенкс	97%
13	013	Наталі Портман	Кіра Найтлі	89%
14	014	Кріс Хемсворт	Кріс Хемсворт	98%
15	015	Ріанна	Ріанна	94%
16	016	Бейонсе	Бейонсе	99%
17	017	Шарліз Терон	Шарліз Терон	95%
18	018	Метт Деймон	Метт Деймон	97%
19	019	Емілія Кларк	Емілія Кларк	93%
20	020	Марк Руффало	Марк Руффало	95%

З таблиці видно, що більшість передбачень є точними, з високим відсотком схожості, що свідчить про гарну здатність моделі диференціювати між різними особами. Відсотки схожості для успішних класифікацій знаходяться у діапазоні від 93% до 99%, що є дуже високим показником.

Деякі помилки в передбаченнях, наприклад, коли Анджеліна Джолі була помилково класифікована як Скарлетт Йоханссон із схожістю в 85%, або коли Раян Гослінг був помилково ідентифікований як Раян Рейнольдс з 87% схожості, вказують на потенційні виклики в розрізненні схожих облич. Ці помилки можуть бути викликані обмеженнями у векторах ознак, які використовувалися для тренування моделі, або неідеальним налаштуванням параметрів ядра SVM [25].

Для кращого розуміння процесу навчання та застосування класифікатора на практиці, розглянемо приклад використання SVM на тренувальних даних. Спочатку, класифікатор було навчено на основі приблизно 20 зображень кожного для Вілла Феррелла, Чада Сміта та Джиммі Фаллона (рис. 2.12).

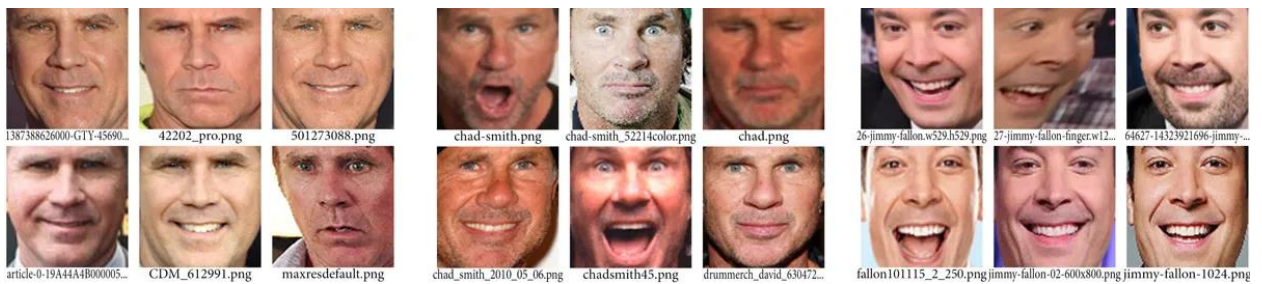


Рисунок 2.12 – Приклад тестових даних [7]

Далі, класифікатор було застосовано до кожного кадру відомого відео з YouTube, де Вілл Феррелл та Чад Сміт імітують один одного на шоу Джиммі Фаллона. Приклад кадру розпізнавання наведено на рисунку 2.13.



Рисунок 2.13 – Визначення облич за допомогою SVM [7]

Результати розпізнавання показують, що класифікатор ідентифікував обличчя Вілла Феррелла з точністю 86%, Чада Сміта з точністю 84%, а Джиммі Фаллона – з точністю 86%. Це свідчить про високу здатність моделі розрізняти навіть дуже схожі обличчя, як-от Феррелла та Сміта, а також точно ідентифікувати обличчя ведучого, Джиммі Фаллона. Така точність ідентифікації демонструє ефективність застосування алгоритму SVM у задачах розпізнавання облич у відео в реальному часі.

3 КОМП'ЮТЕРНА МОДЕЛЬ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи необхідно розробити безпосередньо систему розпізнавання, налаштувати базу даних та сховище тренувальних даних, зробити графічний інтерфейс для користувачів програми. Проектування ефективної комп'ютерної моделі для системи розпізнавання облич у відеопотоці вимагає вибору такого програмного середовища, яке б не тільки забезпечувало високу продуктивність і гнучкість у роботі з великими обсягами даних, але й дозволяло б легко інтегрувати новітні технології та бібліотеки.

PyCharm було обрано як інтегроване середовище розробки (IDE) через його зручність, багатофункціональність і вбудовану підтримку Python, що є однією з найбільш популярних мов програмування в галузі машинного навчання та обробки зображень. PyCharm надає розширені можливості для дебагінгу, тестування коду та управління бібліотеками, що важливо для великих та складних проєктів (рис. 3.1) [28].

За основну мову програмування обрано Python, оскільки вона має багатий набір бібліотек та фреймворків для обробки зображень та машинного навчання, таких як OpenCV і CVZone. Python також відомий своєю здатністю до швидкої розробки та високою читабельністю коду, що спрощує колаборацію та масштабування проєктів.

OpenCV є однією з найбільш використовуваних бібліотек для обробки зображень та комп'ютерного зору, яка забезпечує широкий спектр інструментів для реалізації алгоритмів розпізнавання облич у відеопотоках. Її ефективність і широке прийняття у спільноті зробили її ідеальним вибором для впровадження складних обчислювальних завдань, пов'язаних із розпізнаванням облич [29].

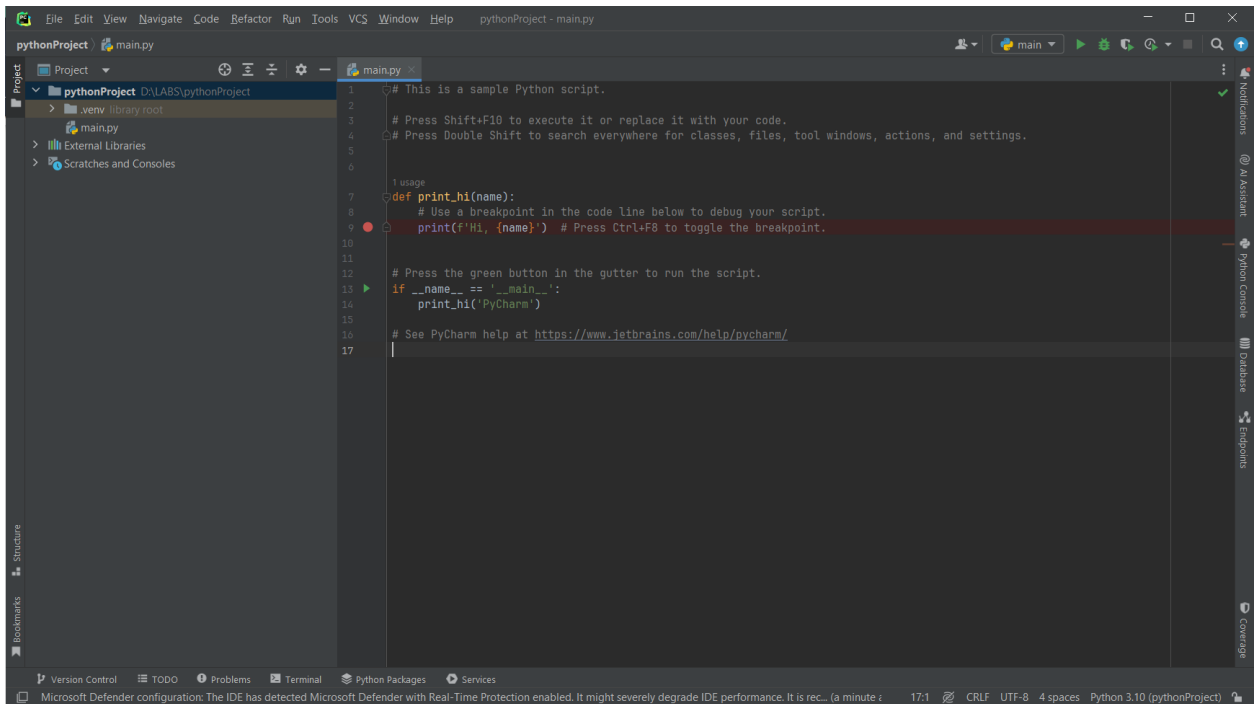


Рисунок 3.1 – Приклад інтерфейсу PyCharm

CVZone використовується для поліпшення процесу детекції та відстеження об'єктів у відео, завдяки своїм оптимізованим інструментам для роботи з кольоровими фільтрами, зонами та ключовими точками на обличчі. Це значно спрощує роботу з точною ідентифікацією ключових точок обличчя і підвищує загальну продуктивність системи [30].

Щодо управління даними в реальному часі та зберігання, Firebase була обрана як платформа для реалізації баз даних та зберігання даних через її масштабованість, надійність і легкість інтеграції з різними сервісами та платформами. Firebase надає рішення для баз даних в реальному часі, що дозволяє системі негайно реагувати на зміни в даних та оптимально керувати даними користувачів (рис. 3.2) [31].

Такий підхід дозволяє створити надійну і масштабовану комп'ютерну модель системи розпізнавання обличчя, здатну ефективно обробляти відеопотоки, точно визначати обличчя та ідентифікувати ключові особливості обличчя у відео.

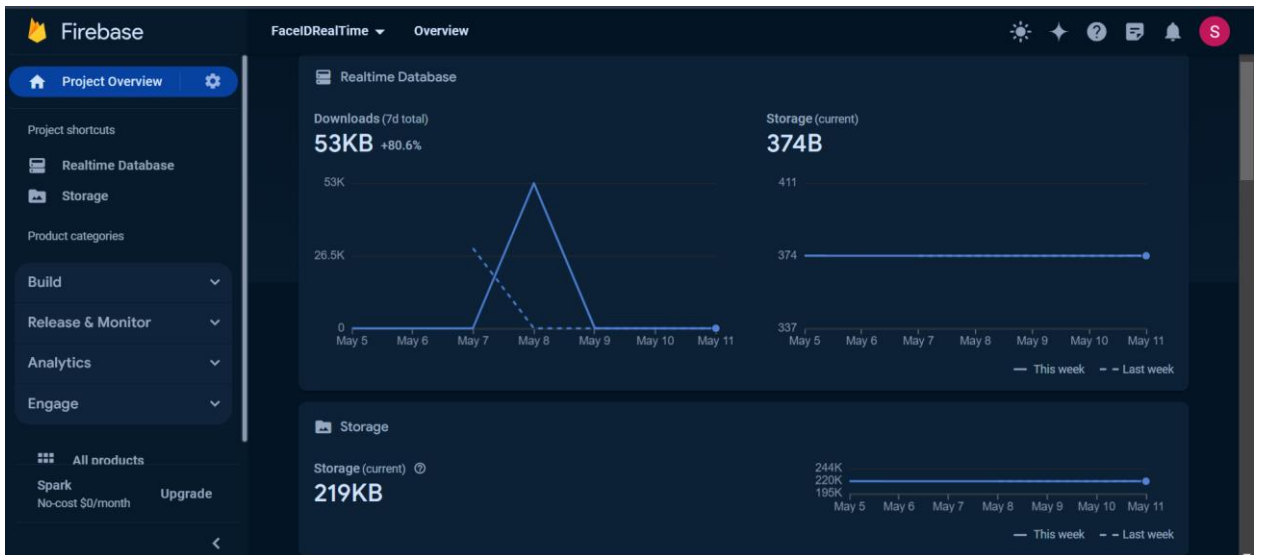


Рисунок 3.2 – Приклад інтерфейсу Firebase

3.2 Програмна реалізація системи розпізнавання облич

Реалізація програми включає декілька основних етапів, які проілюстровано на рисунку 3.3.



Рисунок 3.3 – Етапи розробки застосунку розпізнавання облич у відеопотоці

Система складається з кількох компонентів: модуль розпізнавання облич, модуль кодування облич та модуль завантаження даних у базу даних (рис. 3.4).

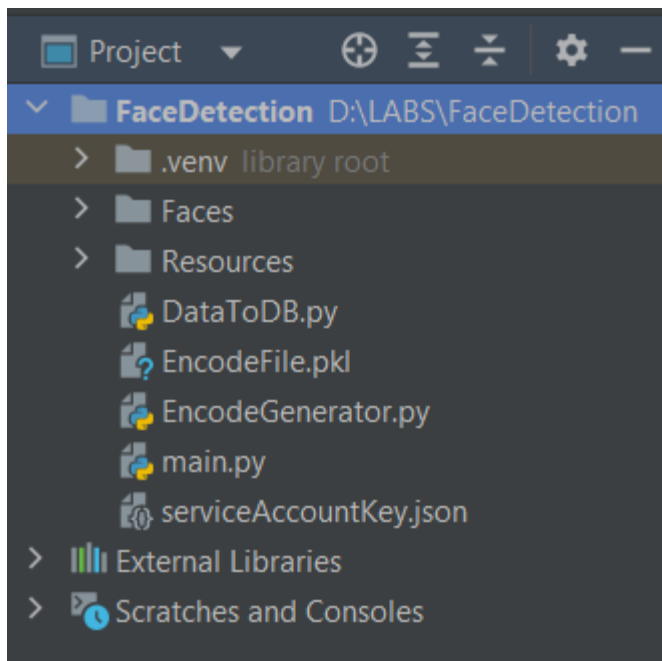


Рисунок 3.4 – Структура проекту

Папка «Faces» має зображення облич осіб розміром 216×216, які зареєстровані у системі. Кожне зображення іменується як індивідуальний id, який наданий особі.

Папка «Resources» має шаблон фонового зображення та п'ять шаблонів для кожного стану розпізнавання системи: «Active», «Profile», «Marked», «Already Marked» і «Unknown Face» (рис. 3.5–3.10).

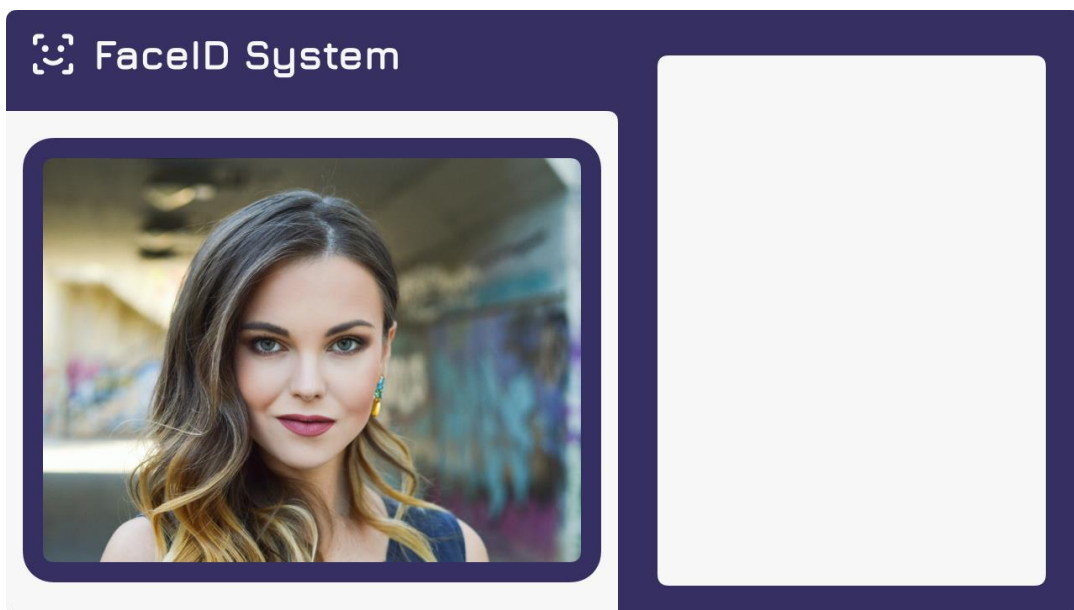


Рисунок 3.5 – Шаблон фону

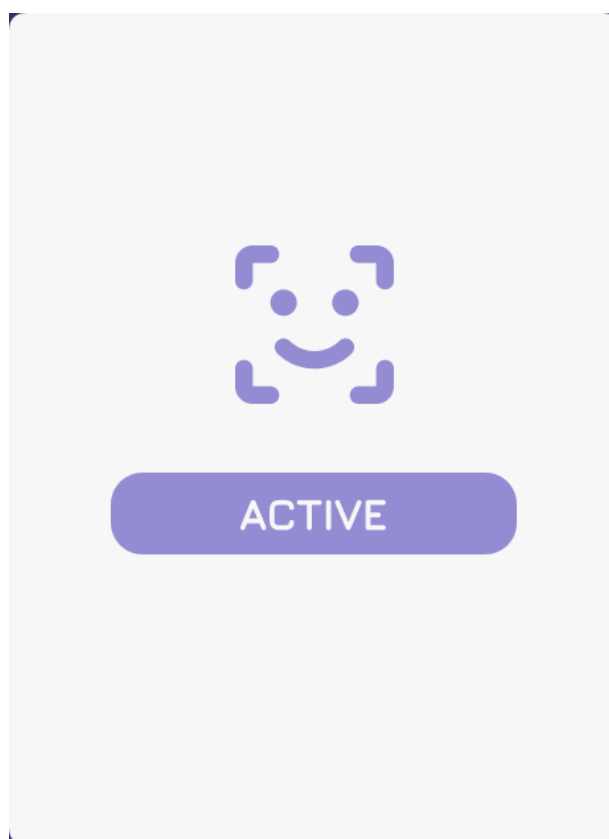


Рисунок 3.6 – Шаблон «Active»

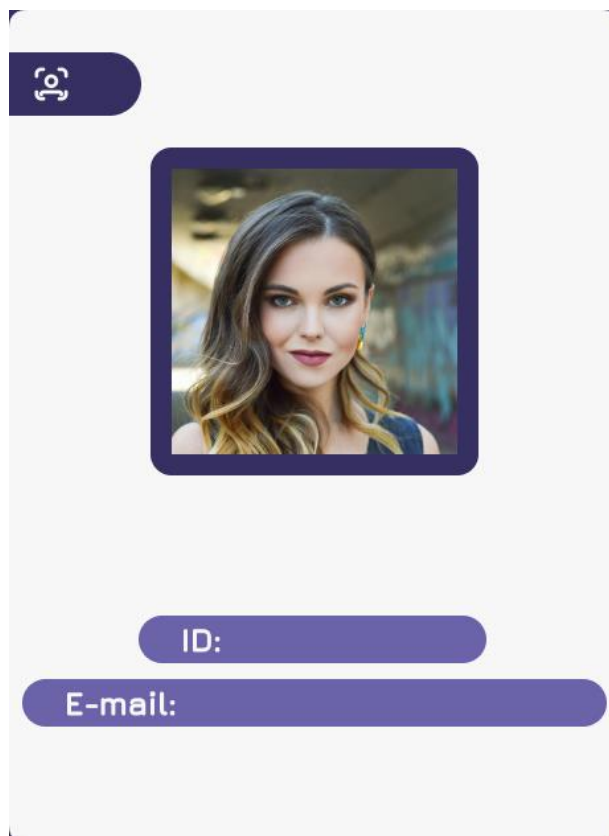


Рисунок 3.7 – Шаблон «Profile»



Рисунок 3.8 – Шаблон «Marked»

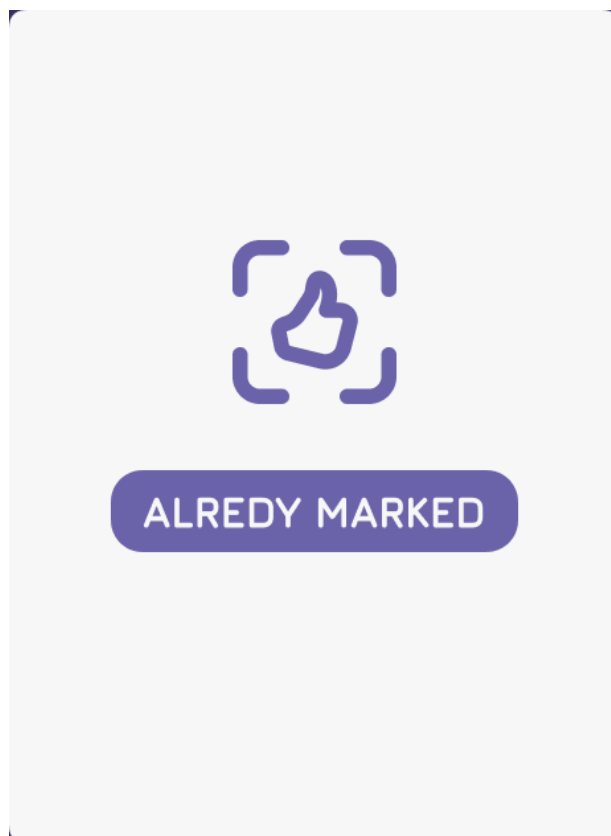


Рисунок 3.9 – Шаблон «Already Marked»



Рисунок 3.10 – Шаблон «Unknown Face»

Для реалізації програмного забезпечення системи розпізнавання облич у відеопотоці необхідно встановити декілька спеціалізованих бібліотек, які забезпечують обробку зображень, аналіз даних та інтерактивну роботу з відео. Серед таких бібліотек є: `smake`, `dlib`, `face_recognition`, `cvzone`, `opencv`, `numpy` [29, 30].

Лістинг 3.1 Приклад встановлення бібліотек через менеджер пакетів `pip`:

```
pip install opencv-python
```

Лістинг 3.2 Приклад встановлення бібліотек із джерел:

```
pip install git+https://github.com/opencv/opencv-python.git
```

Список усіх встановлених бібліотек наведено на рисунку 3.11 та 3.12.

Package	Version	Latest version
CacheControl	0.14.0	0.14.0
PyJWT	2.8.0	2.8.0
cachetools	5.3.3	5.3.3
certifi	2024.2.2	2024.2.2
cffi	1.16.0	1.16.0
charset-normalizer	3.3.2	3.3.2
click	8.1.7	8.1.7
cmake	3.29.2	▲ 3.29.3
colorama	0.4.6	0.4.6
cryptography	42.0.5	▲ 42.0.7
cvzone	1.6.1	1.6.1
dlib	19.22.99	▲ 19.24.4
face-recognition	1.3.0	1.3.0
face-recognition-models	0.3.0	0.3.0
firebase-admin	6.5.0	6.5.0
google-api-core	2.18.0	▲ 2.19.0
google-api-python-client	2.127.0	▲ 2.129.0
google-auth	2.29.0	2.29.0
google-auth-httplib2	0.2.0	0.2.0
google-cloud-core	2.4.1	2.4.1
google-cloud-firestore	2.16.0	2.16.0
google-cloud-storage	2.16.0	2.16.0
google-crc32c	1.5.0	1.5.0
google-resumable-media	2.7.0	2.7.0
googleapis-common-protos	1.63.0	1.63.0
grpcio	1.62.2	▲ 1.63.0
grpcio-status	1.62.2	▲ 1.63.0
httplib2	0.22.0	0.22.0
idna	3.7	3.7

Рисунок 3.11 – Список встановлених бібліотек

imageio	2.34.1	2.34.1
lazy_loader	0.4	0.4
msgpack	1.0.8	1.0.8
networkx	3.3	3.3
numpy	1.26.0rc1	▲ 1.26.4
opencv-python	4.9.0.80	4.9.0.80
packaging	24.0	24.0
pillow	10.3.0	10.3.0
pip	24.0	24.0
proto-plus	1.23.0	1.23.0
protobuf	4.25.3	▲ 5.26.1
pyasn1	0.6.0	0.6.0
pyasn1_modules	0.4.0	0.4.0
pycparser	2.22	2.22
pyparsing	3.1.2	3.1.2
requests	2.31.0	2.31.0
rsa	4.9	4.9
scikit-image	0.23.2	0.23.2
scipy	1.13.0	1.13.0
setuptools	68.2.0	▲ 69.5.1
tiffio	2024.4.24	▲ 2024.5.10
uritemplate	4.1.1	4.1.1
urllib3	2.2.1	2.2.1
wheel	0.41.2	▲ 0.43.0

Рисунок 3.12 – Продовження списку встановлених бібліотек

Для роботи з базами даних і зберіганням даних у реальному часі використовується Firebase. Для інтеграції Firebase із Python проектом, необхідно встановити Firebase SDK [31].

Лістинг 3.3 Приклад встановлення Firebase SDK через менеджер пакетів `pip`:

```
pip install firebase-admin
```

Лістинг 3.4 Ініціалізація з'єднання з Firebase, використовуючи облікові дані із JSON-файлу «`serviceAccountKey.json`»:

```
cred = credentials.Certificate("serviceAccountKey.json")  
firebase_admin.initialize_app(cred, {  
    'databaseURL': "Database URL",  
    'storageBucket': "Storage URL"  
})
```

Після успішного з'єднання з Firebase можна завантажувати необхідні дані у базу даних та сховище даних (рис. 3.13 і 3.14).

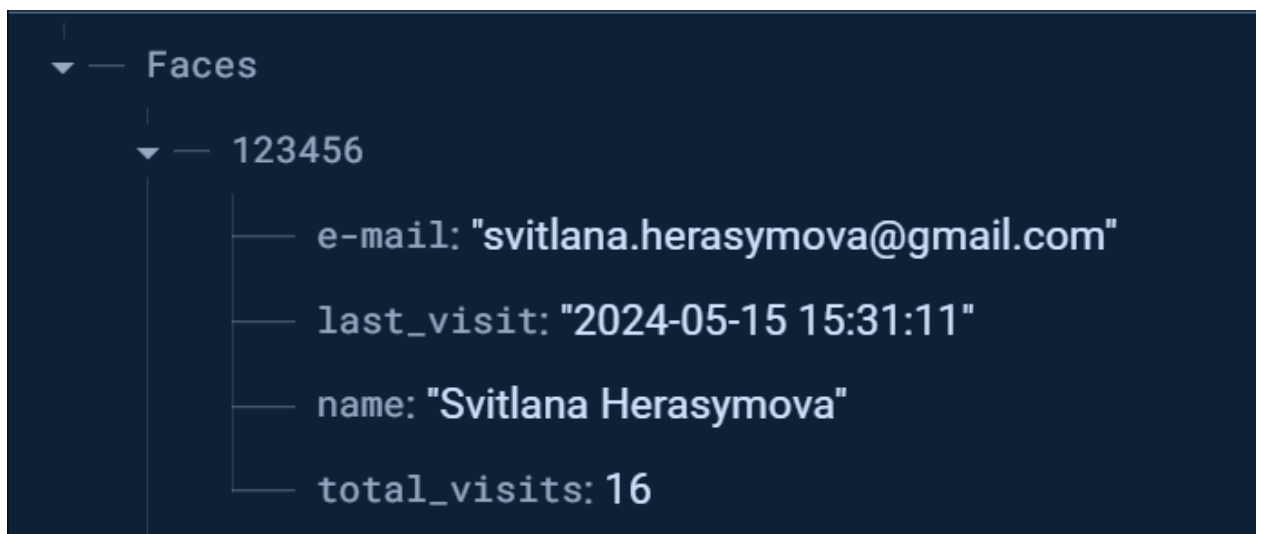


Рисунок 3.13 – Приклад завантажених даних у Realtime Database




<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	 123456.png	56.56 KB	image/png	Apr 30, 2024
<input type="checkbox"/>	 852741.png	84.5 KB	image/png	Apr 30, 2024
<input type="checkbox"/>	 963852.png	77.56 KB	image/png	Apr 30, 2024

Рисунок 3.14 – Приклад завантажених зображень у Storage

Після встановлення всіх необхідних бібліотек рекомендується перевірити їх наявність та версії, щоб забезпечити правильну інтеграцію у проект.

Лістинг 3.5 Перевірка встановлених бібліотек та їх версій:

```
pip list
```

Ця команда виведе список усіх встановлених бібліотек з їхніми версіями, дозволяючи переконатися, що все необхідне було встановлено коректно.

3.3 Інструкція користувача

Програма призначена для онлайн ідентифікації осіб у відеопотоці, використовуючи технології машинного навчання. Інтерфейс програми був розроблений таким чином, щоб взаємодія з користувачем була простою та інтуїтивною.

Реєстрація в системі ідентифікації осіб налаштована так, що кожен користувач може зареєструватися лише один раз на добу. Це обмеження встановлене для оптимізації процесу управління даними та забезпечення точності роботи системи, запобігаючи надмірному накопиченню вхідних даних протягом короткого періоду часу.

Проте, цей параметр є змінним і може бути адаптований відповідно до специфічних потреб використання системи. Залежно від контексту застосування та оперативних вимог, частота реєстрації може бути збільшена або зменшена. Наприклад, у сценаріях, де необхідно частіше відслідковування активності користувачів у чутливих зонах, керівництво може вирішити дозволити кілька реєстрацій на добу.

Після запуску, користувачу буде доступне основне вікно системи, яке містить інтерфейс вебкамери для зображення в реальному часі та панель статусів системи (рис. 3.15).

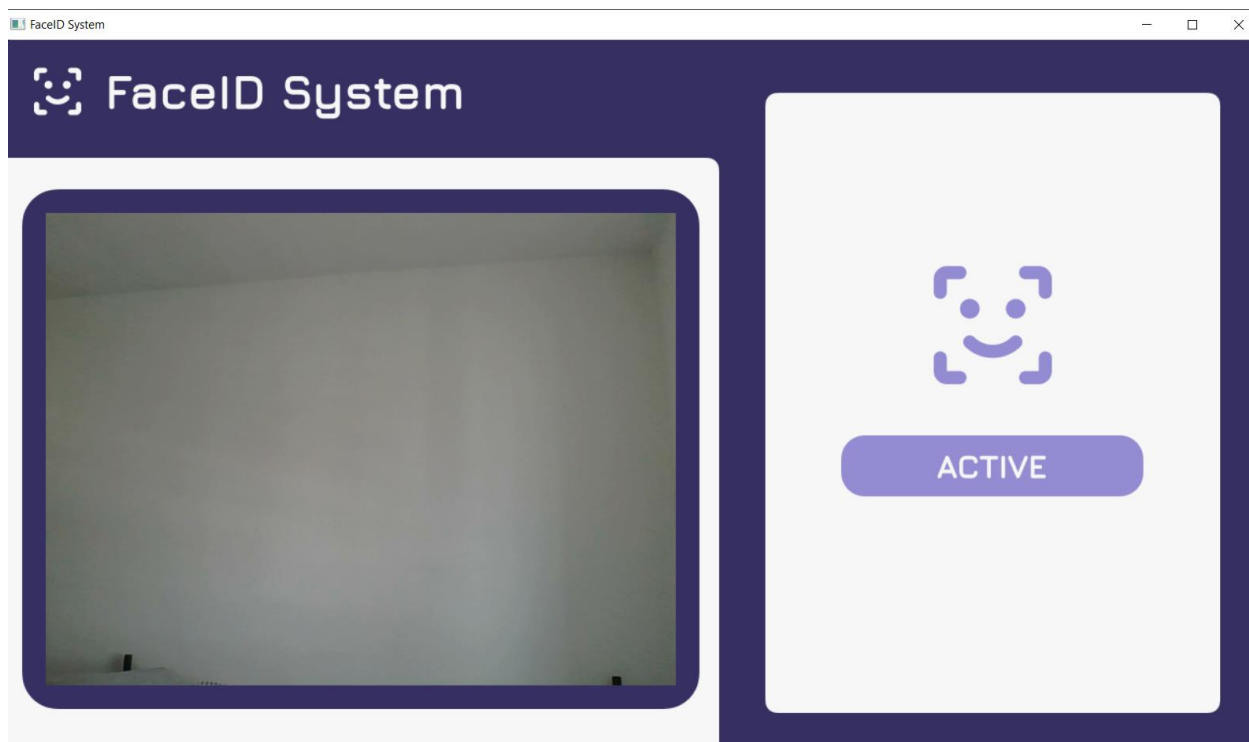


Рисунок 3.15 – Головне вікно системи

Система розпізнавання облич передбачає п'ять основних станів оперативного режиму: «Active», «Profile», «Marked», «Already Marked» та «Unknown Face». Кожен з цих станів відіграє важливу роль у процесі ідентифікації та управлінні даними облич.

Стан «Active» вказує на те, що система знаходиться в активному режимі спостереження та аналізу, здійснюючи безперервне сканування та

розпізнавання облич. Активація цього стану забезпечує, що система налаштована на виявлення нових облич або верифікацію осіб зі збереженої бази даних (рис. 3.16).

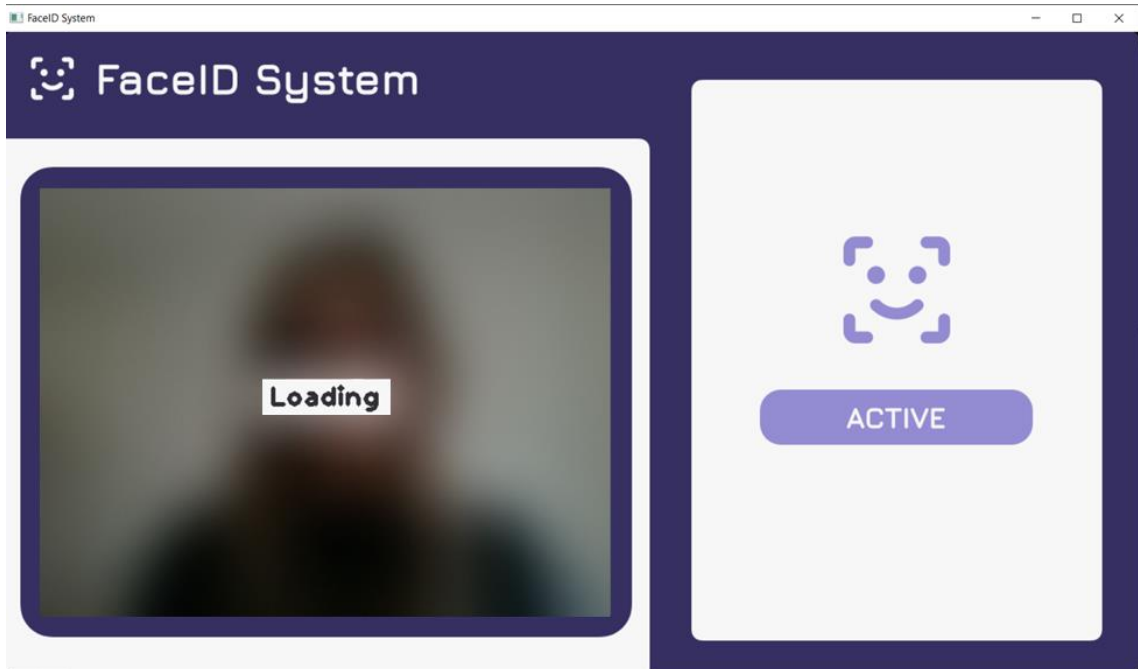


Рисунок 3.16 – Демонстрація стану системи «Active»

Під час процесу ідентифікації осіб програма здійснює декілька дій для інформування користувача про поточний стан обробки. Коли система виявляє обличчя в потоці відео, на екрані з'являється надпис «Loading...», що свідчить про процес аналізу та обробки даних. Це індикатор того, що програма виконує необхідні розрахунки для визначення ідентичності особи.

При успішному розпізнаванні особи система переходить у стан «Profile», автоматично генеруючи картку користувача, яка містить необхідну інформацію, включаючи фото, загальну кількість входжень у систему, ім'я, ідентифікаційний номер та електронна адреса. Це дозволяє користувачам системи отримати повний доступ до інформації про особу для подальших дій або записів (рис. 3.17).

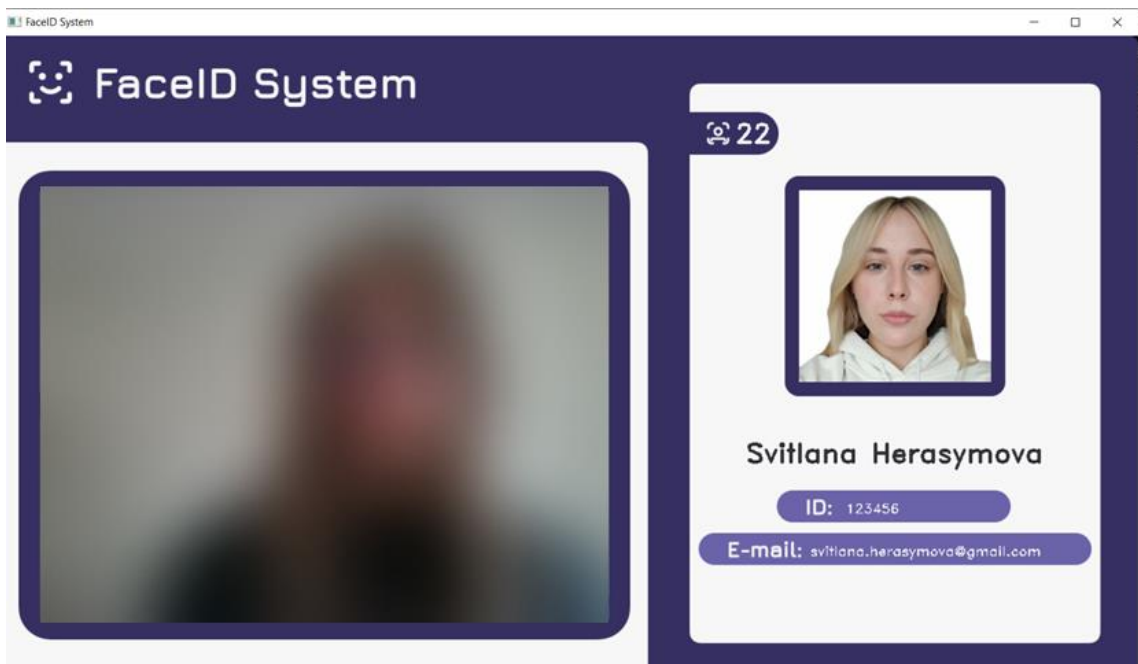


Рисунок 3.17 – Демонстрація стану системи «Profile»

Індикатори «Marked» та «Already Marked» вказують на статус розпізнаної особи у системі. «Already Marked» відображається, якщо особу вже було ідентифіковано і відмічено у базі даних, в той час як «Marked» з'являється, коли особа щойно була ідентифікована та додана до бази (рис. 3.18 і 3.19).

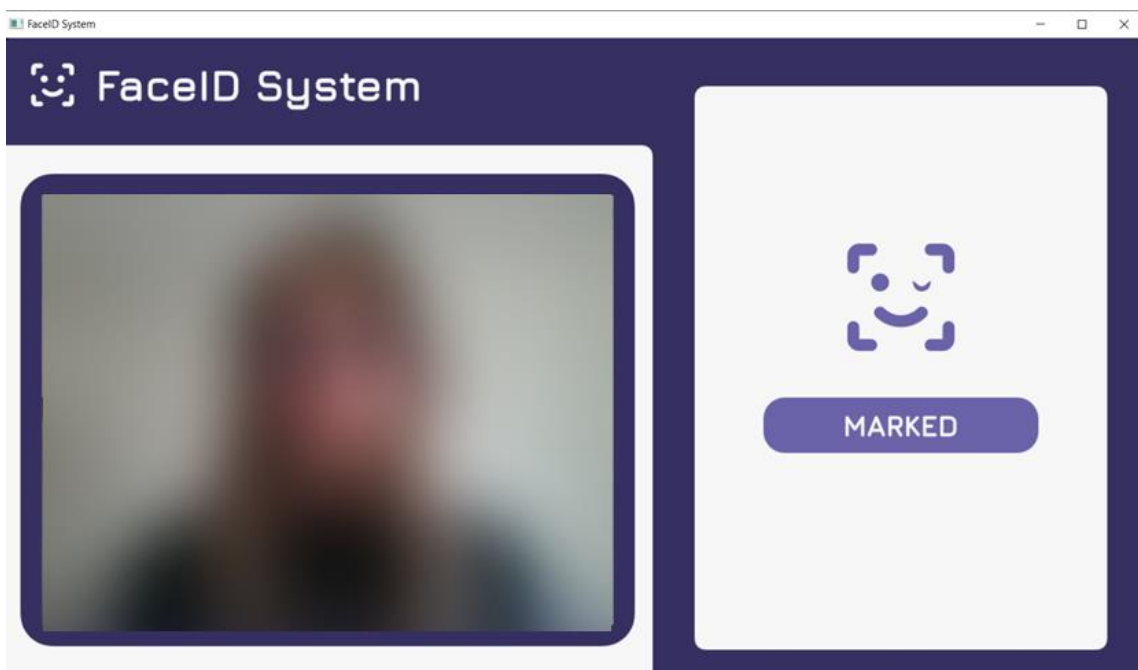


Рисунок 3.18 – Демонстрація стану системи «Marked»

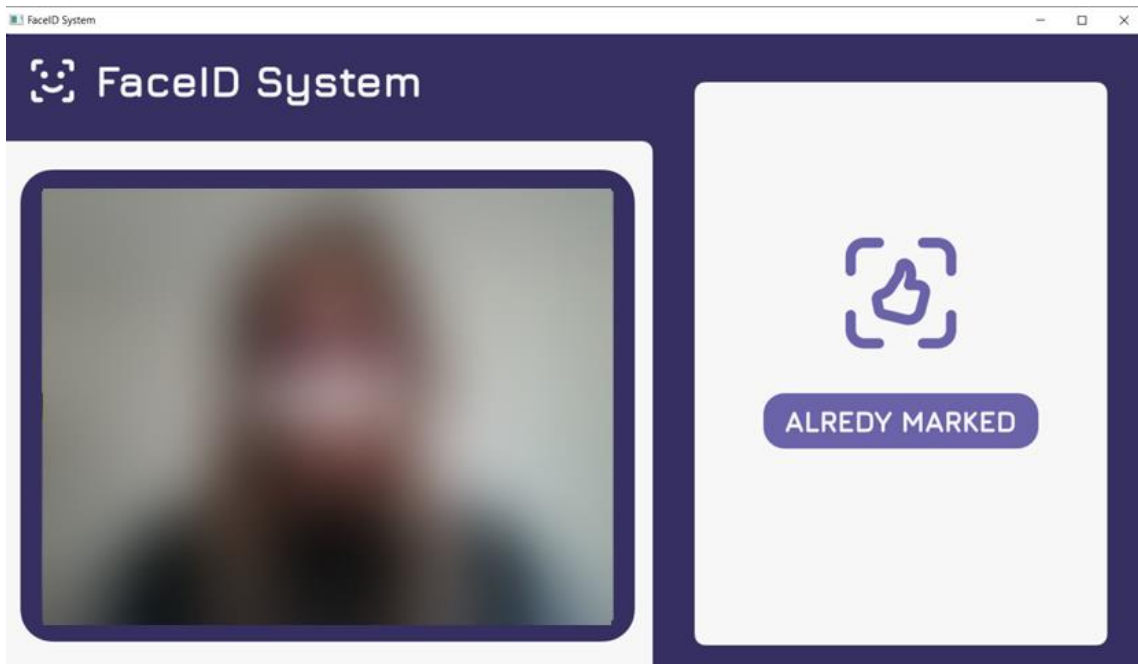


Рисунок 3.19 – Демонстрація стану системи «Already Marked»

Стан «Unknown Face» активізується, коли система зіткнулася з обличчям, яке не може бути співставлене з жодним існуючим записом у базі даних. Це викликає потребу в додаткових діях з боку адміністратора системи, наприклад, ручній перевірці або наданні можливості для реєстрації нового користувача. Стан «Unknown Face» показано на рисунку 3.20.

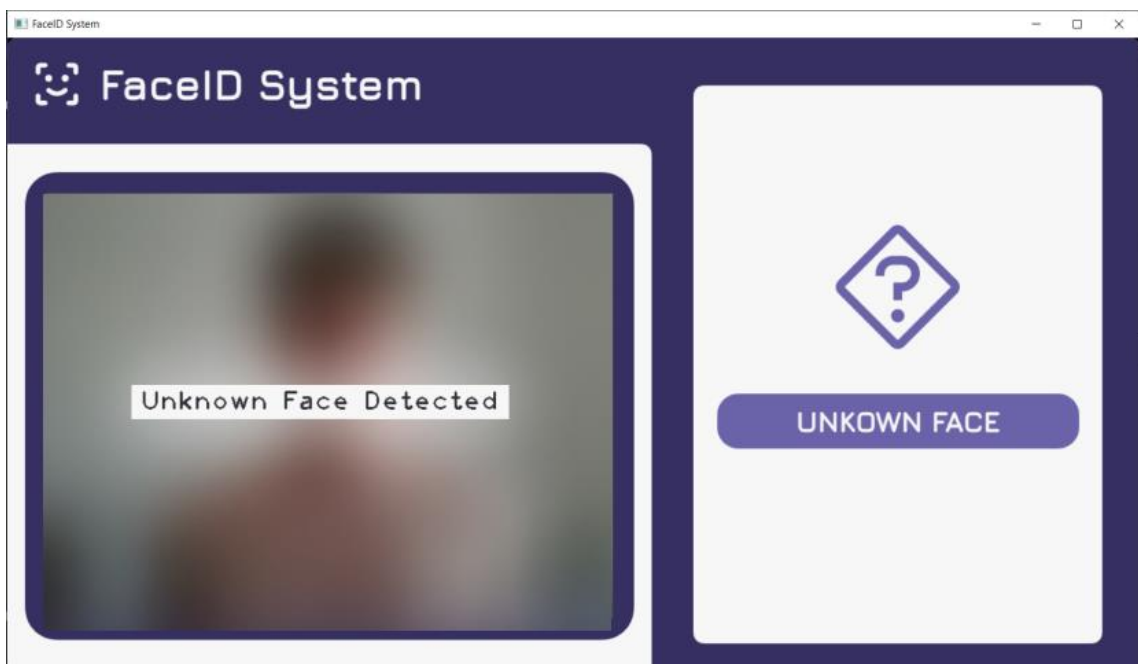


Рисунок 3.20 – Демонстрація стану системи «Unknown Face»

У випадку, коли обличчя, яке з'являється в кадрі, не відповідає жодному зі збережених у базі даних профілів, система виводить повідомлення «Unknown Face Detected». Це означає, що система зіткнулася з обличчям, яке не може бути ідентифіковано як відоме.

Кожен з цих станів формує інтегрований механізм взаємодії системи, спрямований на забезпечення ефективності, безпеки та зручності використання, відповідно до політик доступу та ідентифікації користувачів.

Після закінчення використання програми, необхідно натиснути на хрестик у правому верхньому куті або «Q» на клавіатурі.

Для зберігання інформації про користувачів у Firebase Realtime Database необхідно скористатися графічним інтерфейсом користувача, який дозволяє легко вводити дані та перевіряти їх правильність (рис. 3.21).




The image shows a web browser window titled "FaceID System". The main content area has a dark blue background with the text "FaceID System" in white. Below this, there is a form with five input fields, each with a label to its left: "User ID", "Name", "E-mail", "Total Visits", and "Last Visit (Optional)". At the bottom of the form, there are two buttons: "Add Data" and "Clear".

Рисунок 3.21 – Вікно для додавання даних у БД

Після запуску програми буде доступно вікно з наступними елементами:

- поле «User ID» у яке треба ввести унікальний ідентифікатор користувача (від 3 до 10 цифр);
- поле «Name» у яке треба ввести ім'я користувача;
- поле «Email» у яке треба ввести адресу електронної пошти користувача (наприклад, example@example.com);
- поле «Total Visits» у яке треба ввести кількість відвідувань користувачем;
- поле введення «Last Visit (optional)» у яке треба ввести дату останнього відвідування у форматі «YYYY-MM-DD HH:MM:SS» (це поле необов'язкове).

Заповніть усі необхідні поля, потім натисніть кнопку «Add Data» для додавання даних до Firebase (рис. 3.22).



The screenshot shows a window titled "FaceID System" with a dark blue background. The title bar includes a feather icon, the text "FaceID System", and standard window controls (minimize, maximize, close). The main content area features the text "FaceID System" in a large, white, sans-serif font. Below this, there is a form with five input fields, each with a label and a value:

User ID	232323
Name	Kate Black
E-mail	kate.black@gmail.com
Total Visits	6
Last Visit (Optional)	2024-04-24 19:03:33

At the bottom of the form, there are two buttons: "Add Data" and "Clear", both with a light blue gradient and white text.

Рисунок 3.22 – Приклад заповнення даними

Якщо дані введені правильно, з'явиться повідомлення про успішне додавання даних.

У випадку помилки введення, з'явиться відповідне повідомлення з поясненням помилки.

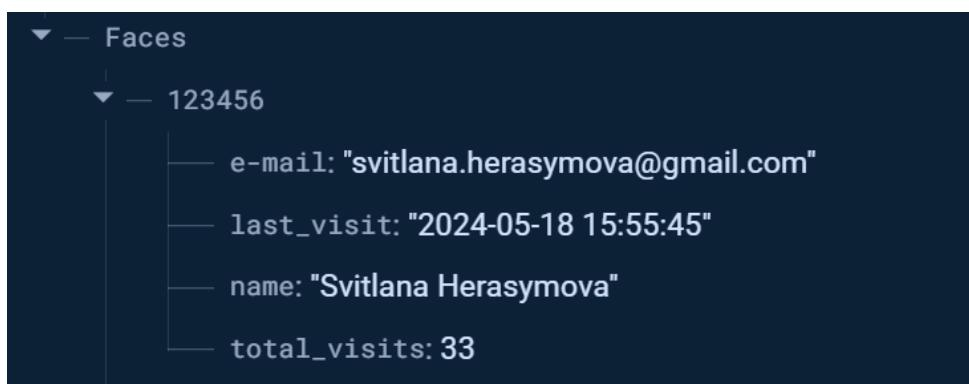
Для очищення всіх полів введення натисніть кнопку «Clear».

Після закінчення використання програми, необхідно натиснути на хрестик у правому верхньому куті.

3.4 Тестування розробленої моделі

Процес тестування розробленої моделі розпізнавання облич у відеопотоці є ключовим етапом, що дозволяє перевірити ефективність та надійність системи у реальних умовах.

Для наочності під час тестування була реалізована можливість реєстрації облич кожну хвилину. Цей параметр був обраний як стандартний для проведення первісних тестів, проте в залежності від специфічних потреб і вимог до системи його можна змінювати, адаптуючи частоту реєстрації до необхідної реакції системи. Приклад реєстрації особи через хвилину наведено на рисунках 3.23 та 3.24.



```
▼ — Faces
  ▼ — 123456
    — e-mail: "svitlana.herasymova@gmail.com"
    — last_visit: "2024-05-18 15:55:45"
    — name: "Svitlana Herasymova"
    — total_visits: 33
```

Рисунок 3.23 – Дані у БД до повторної реєстрації

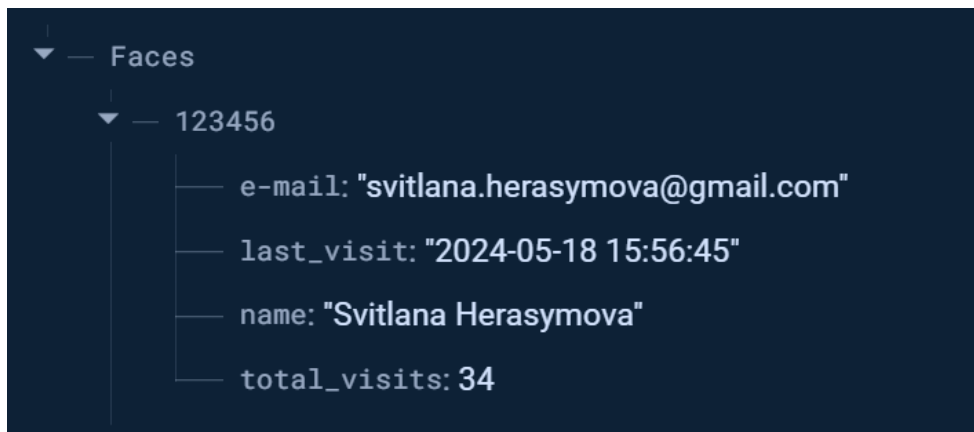


Рисунок 3.24 – Оновлені дані у БД після повторної реєстрації через хвилину

Тестування системи проводилося у різних умовах, щоб забезпечити її універсальність і здатність адаптуватися до змінних зовнішніх факторів.

Під час тестування системи розпізнавання облич у відеопотоці при нормальних умовах освітлення та стандартному зовнішньому вигляді, результати виявились позитивними, як і очікувалось (рис. 3.25).

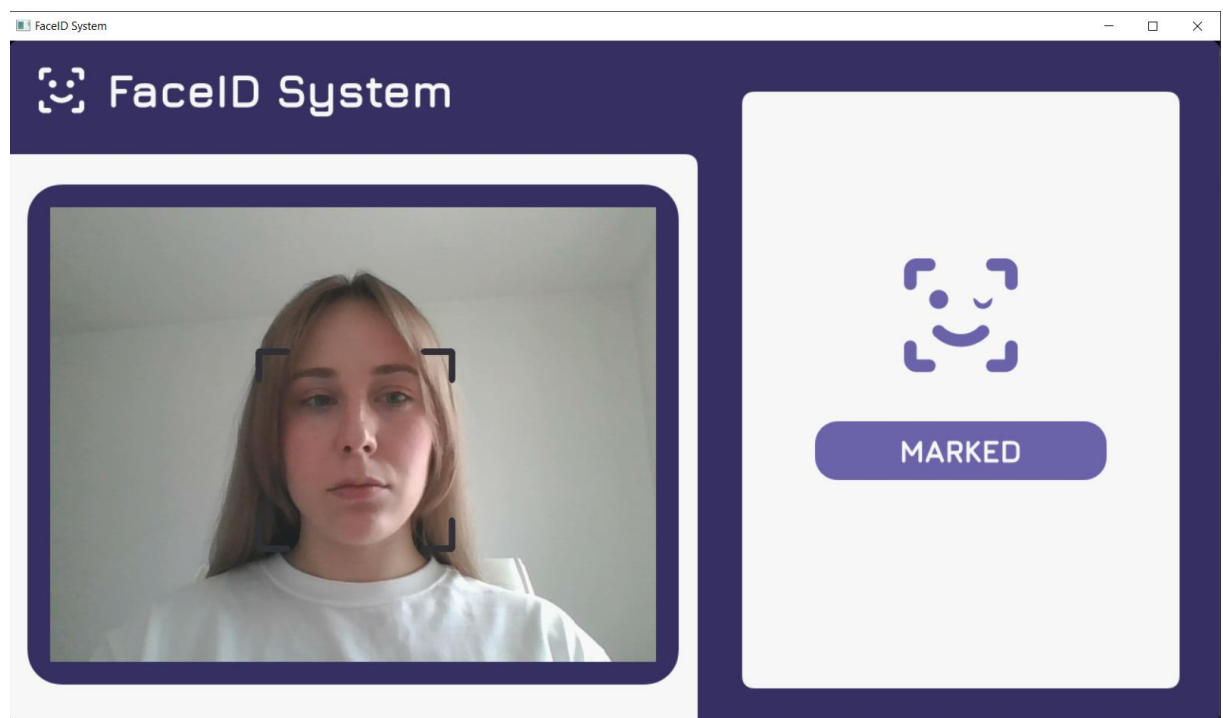


Рисунок 3.25 – Тестування при нормальних умовах

Система демонструвала високу точність ідентифікації осіб, навіть коли вони вносили зміни в свій зовнішній вигляд, такі як зміна зачіски,

використання прозорих окулярів для зору, носіння темних сонцезахисних окулярів чи головного убору. Приклади показано на рисунках 3.26–3.29.

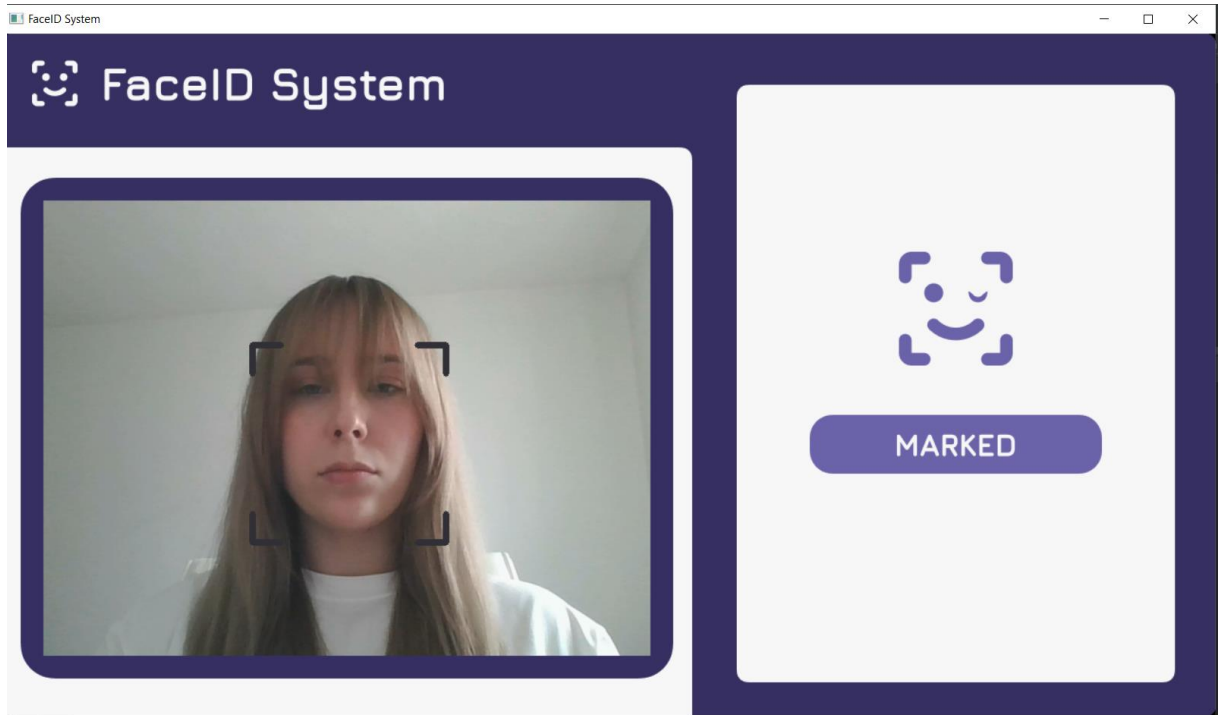


Рисунок 3.26 – Тестування при зміні зачіски

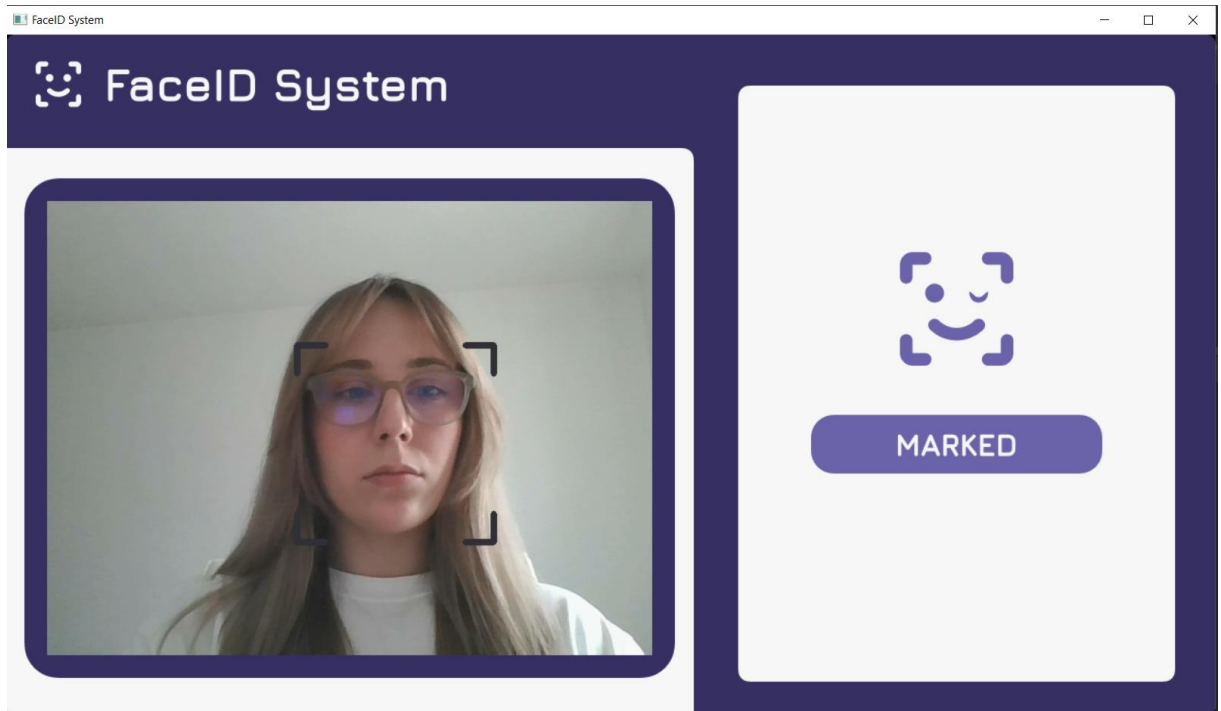


Рисунок 3.27 – Тестування при наявності прозорих окулярів

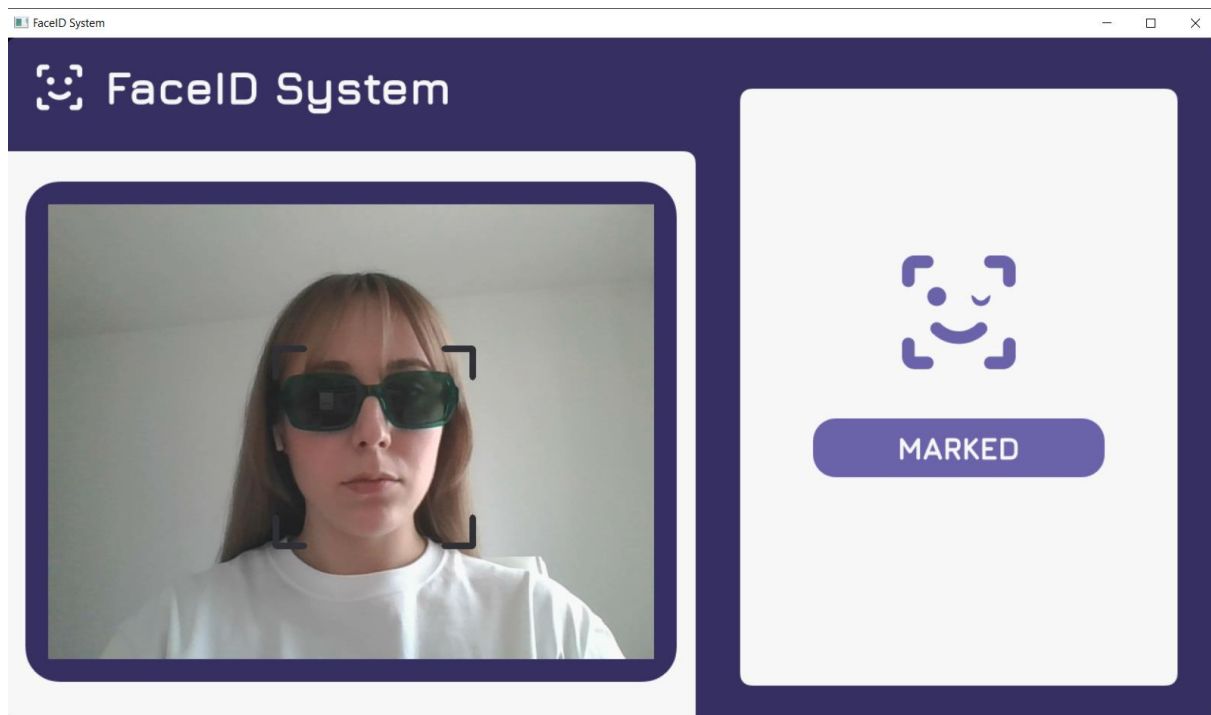


Рисунок 3.28 – Тестування при наявності непрозорих окулярів

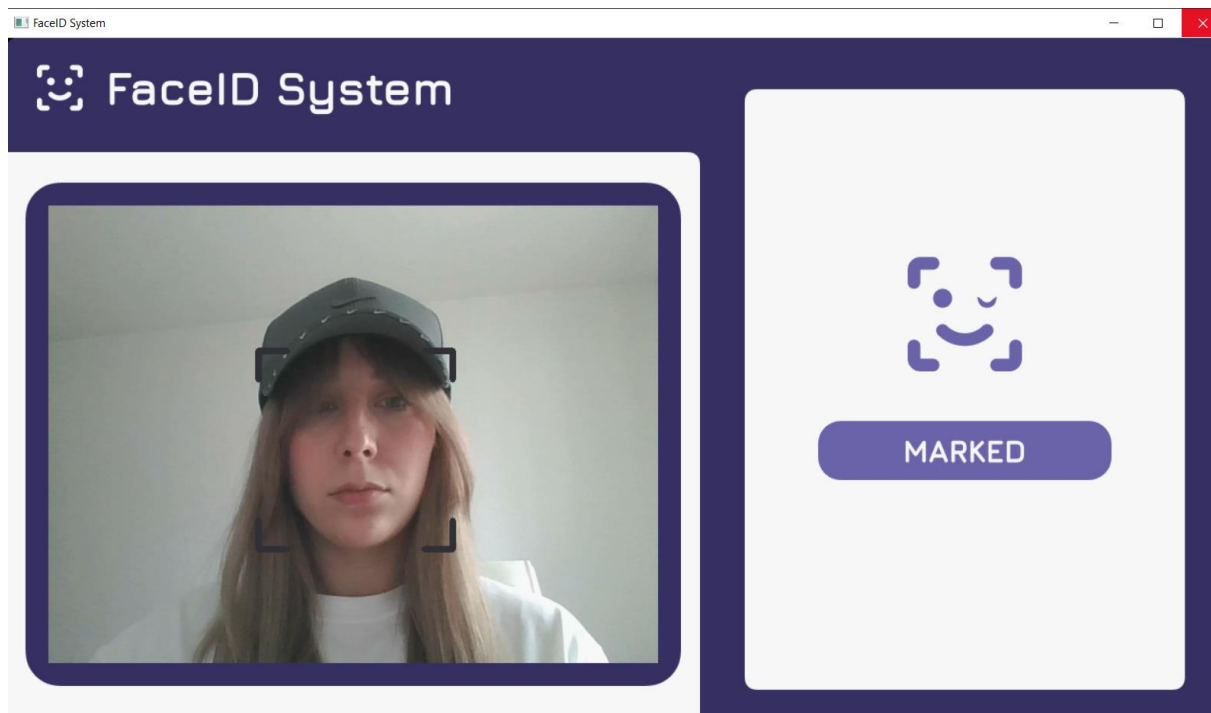


Рисунок 3.29 – Тестування при наявності головного убору

Однак, було виявлено, що коли особа одночасно носила головний убір та темні сонцезахисні окуляри, система не змогла ідентифікувати обличчя, класифікуючи його як «Unknown face» (рис. 3.30).

Це вказує на необхідність подальшого удосконалення алгоритмів системи для кращого розпізнавання обличчя під час одночасного використання кількох елементів, які приховують риси обличчя.

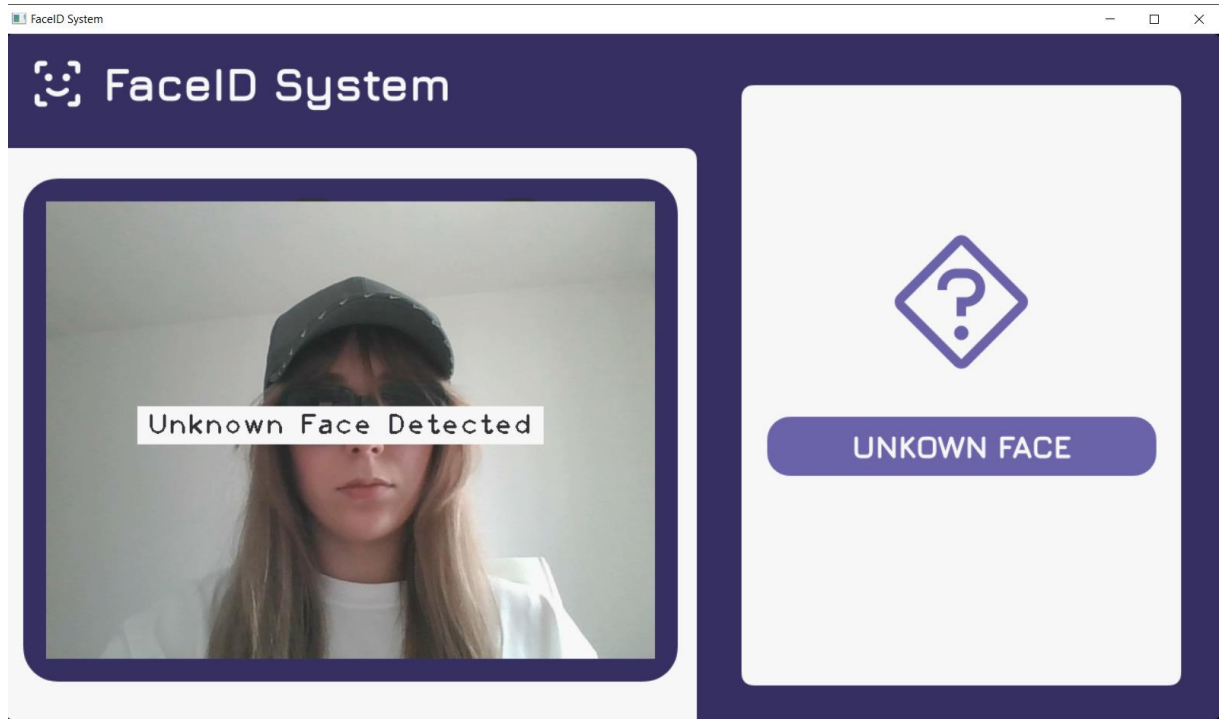


Рисунок 3.30 – Тестування при наявності непрозорих окулярів та головного убору

Тестування системи також проводилося при різних умовах освітленості: від яскравого сонячного світла до темного штучного освітлення. В усіх випадках система показала стабільно позитивні результати, що свідчить про її високу надійність у різноманітних освітлювальних умовах.

Перевірка системи на вулиці під прямими сонячними променями дозволила оцінити її здатність виявляти та ідентифікувати обличчя, незважаючи на потенційне пересвічування частин обличчя та тіні. Система продемонструвала хорошу стабільність, коректно ідентифікуючи обличчя у більшості випадків.

Звичайне денне світло є ідеальними умови для тестування системи, де вона показала найвищу точність і швидкість реакції. Обличчя

ідентифікувались без помилок, підтверджуючи високу ефективність системи у стандартних умовах.

Перевірка при різних рівнях освітленості є фундаментальною для визначення можливостей та обмежень системи. Загалом, система показала високий рівень адаптивності до різних освітлювальних умов, що робить її придатною для використання в різноманітних застосуваннях, від персональної ідентифікації до систем безпеки та моніторингу. Зафіксовані випадки зменшення продуктивності в умовах недостатнього освітлення вказують на потребу подальшої оптимізації алгоритмів, щоб підвищити надійність системи навіть у складних умовах.

Приклади тестових зображень з різним освітленням показано на рисунках 3.31–3.35.

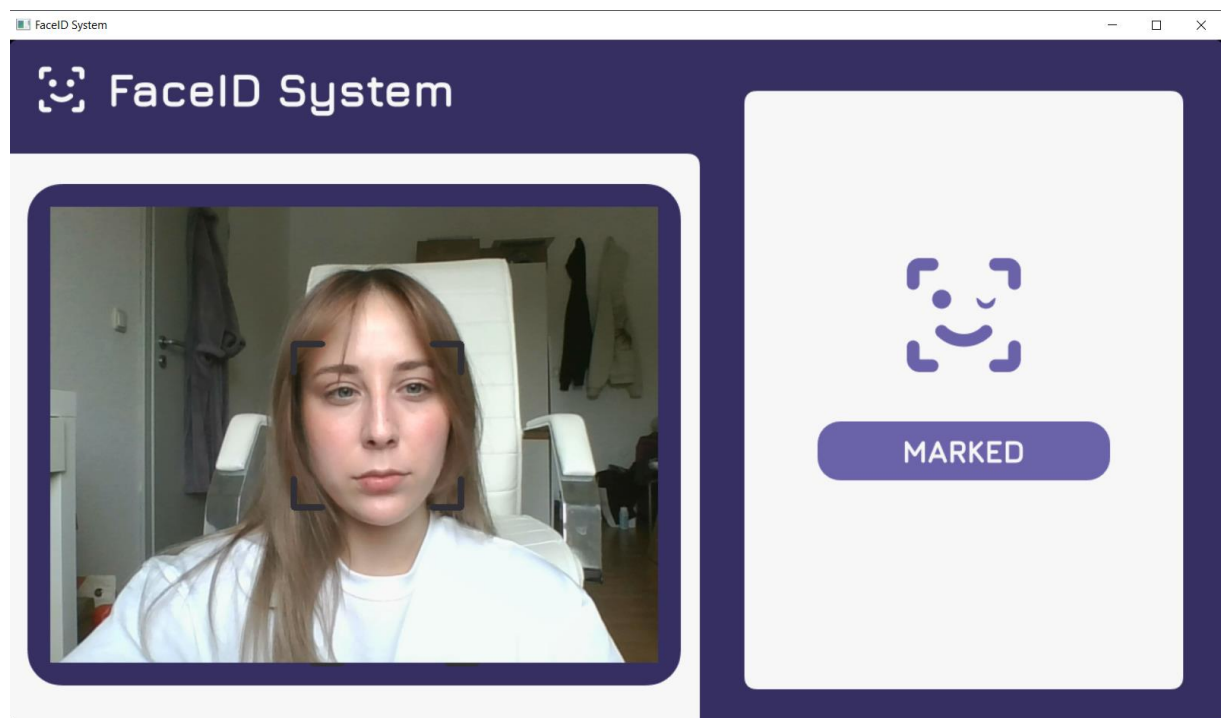


Рисунок 3.31 – Тестування при яскравому освітленні

Також тестування включало роботу системи в умовах слабого освітлення, вечірнього часу та в приміщеннях з недостатнім освітленням. Система змогла адаптуватися до змін умов і продовжувала ефективно

виявляти обличчя, хоча і з невеликим зниженням точності порівняно з оптимальним освітленням (рис. 3.32–3.34).

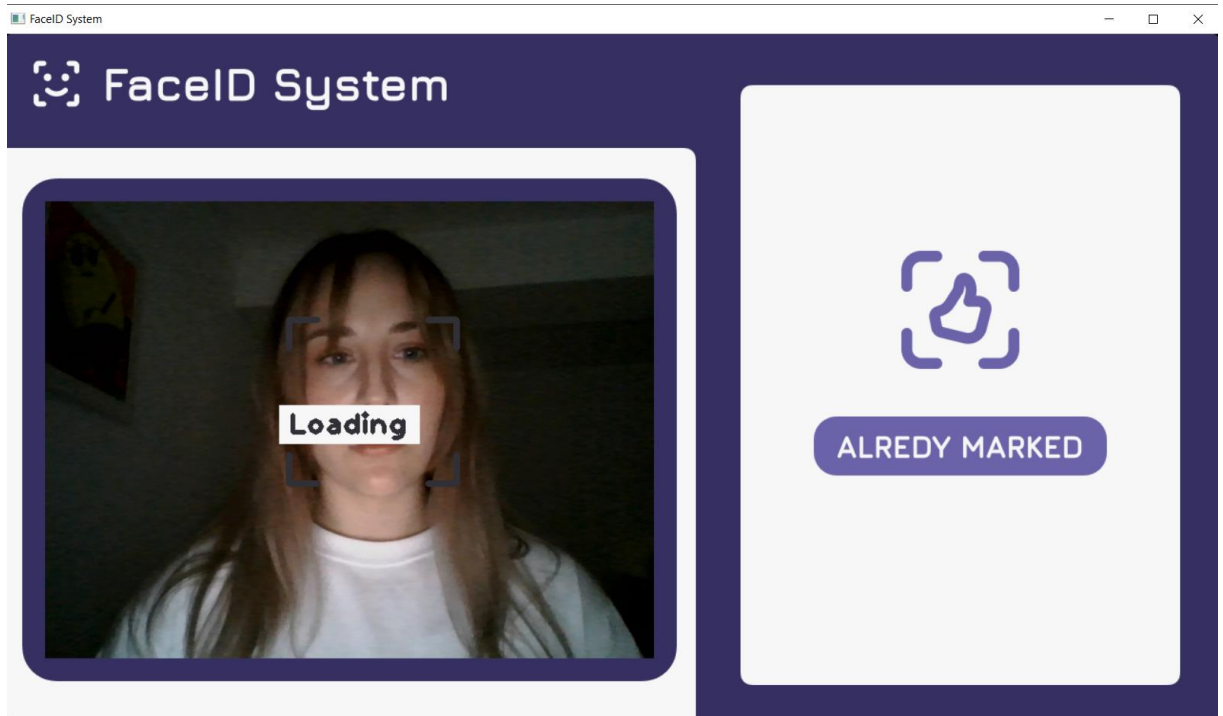


Рисунок 3.32 – Тестування при помірно темному освітленні

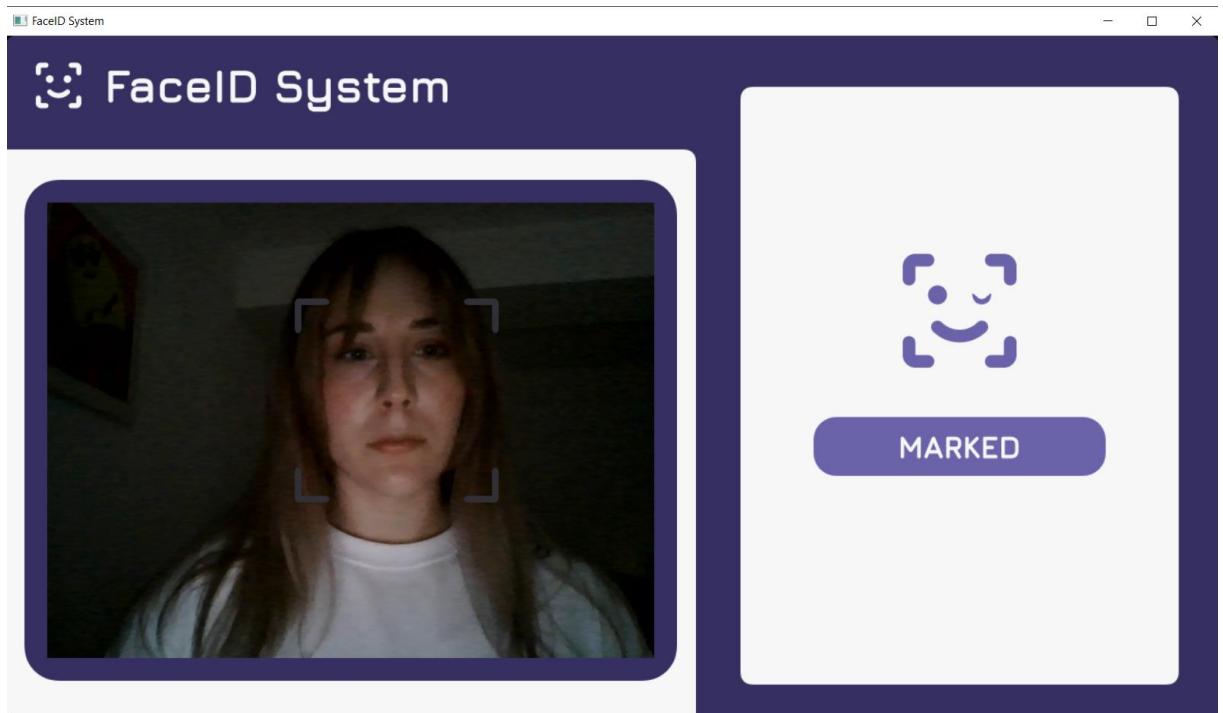


Рисунок 3.33 – Тестування при більш темному освітленні

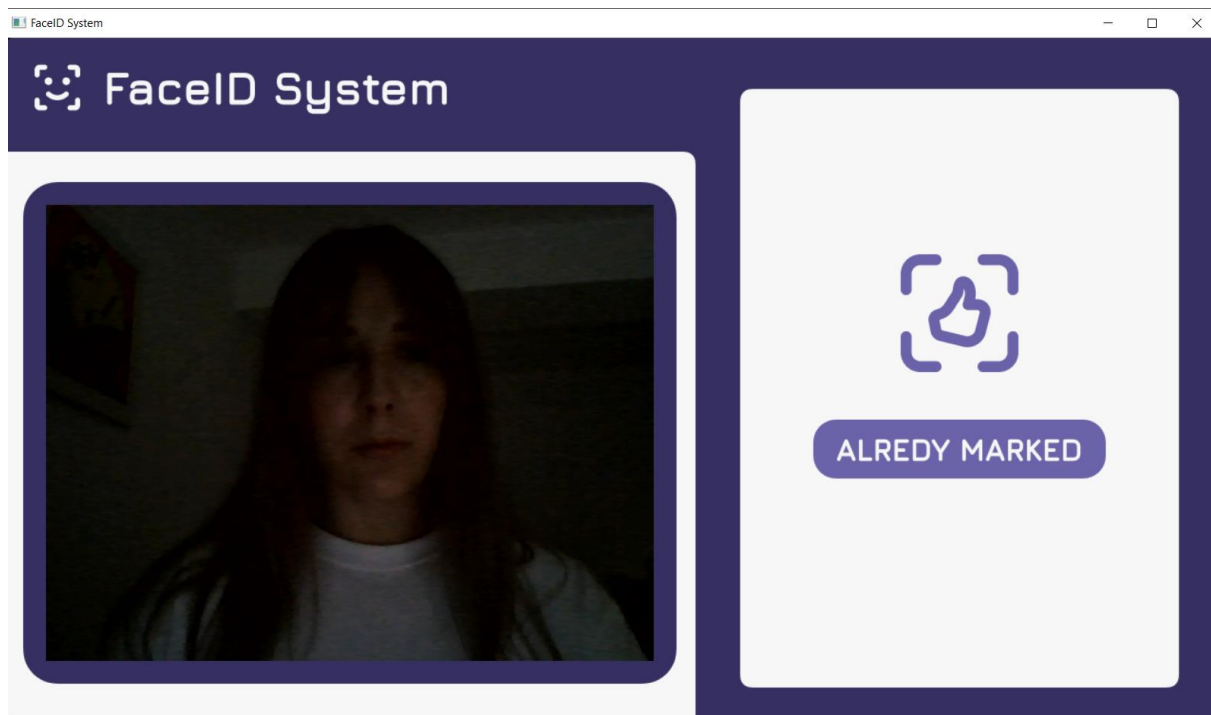


Рисунок 3.34 – Тестування при темному освітленні

Умови з штучним освітленням, такі як офісні приміщення або домашнє освітлення, також були перевірені. Система показала стабільність і надійність, успішно розпізнаючи особи, незалежно від типу ламп (рис. 3.35).

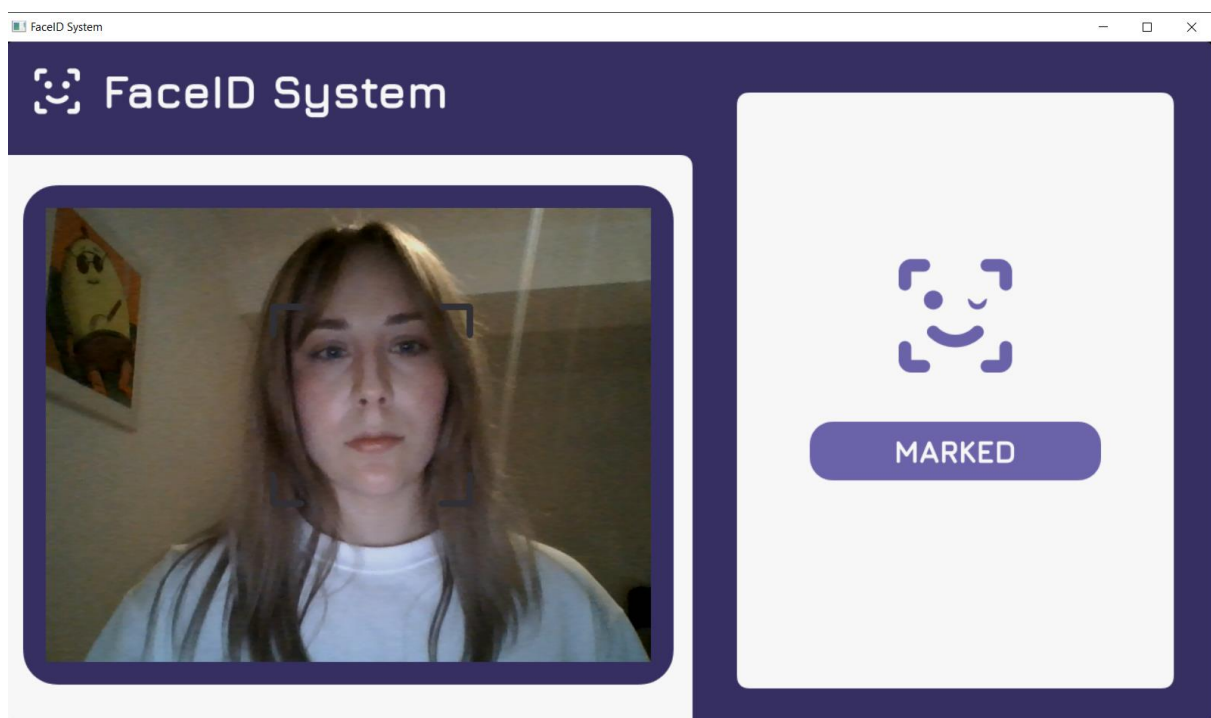


Рисунок 3.35 – Тестування при штучному освітленні

Додатково було проведено тестування системи на особі, яка спочатку не була занесена до бази даних. У цьому випадку система коректно ідентифікувала обличчя як «Unknown face» (рис. 3.36).

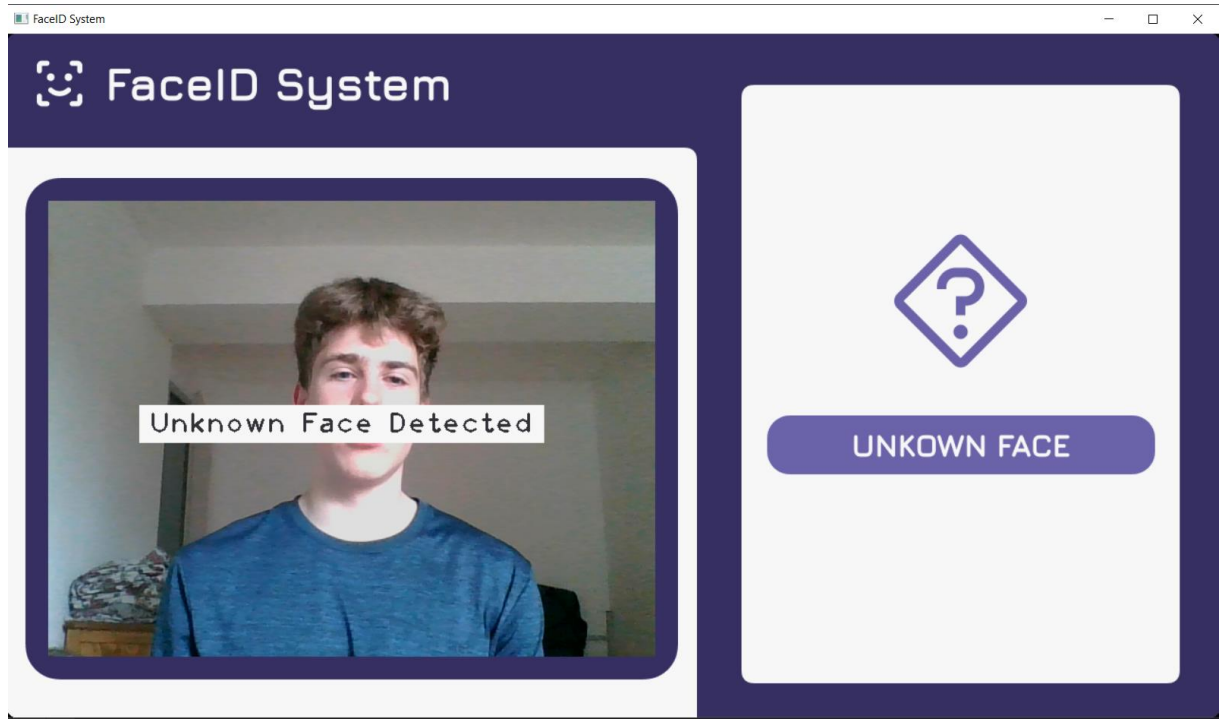


Рисунок 3.36 – Тестування незареєстрованої у системі особи

Стан «Unknown face» у системах розпізнавання облич має важливе значення для забезпечення безпеки та управління доступом. Цей стан використовується для ідентифікації ситуацій, коли особу, що з'являється перед камерою, не можна впізнати або вона відсутня в базі даних. Це особливо корисно у високочутливих застосуваннях, таких як системи контролю доступу в корпоративних або урядових установах, де необхідно точно відрізнати авторизованих користувачів від неавторизованих.

Використання стану «Unknown face» дозволяє системі запускати спеціальні процедури безпеки або сповіщення, коли з'являється особа, яка не відповідає жодному профілю у базі даних. Це може включати відправлення сповіщень охоронним службам, активацію додаткових заходів перевірки або блокування доступу до певних ресурсів чи територій.

Після того, як інформація про особу була додана до бази даних, система успішно ідентифікувала цю особу, що демонструє ефективність механізму навчання та оновлення даних у системі (рис. 3.37).

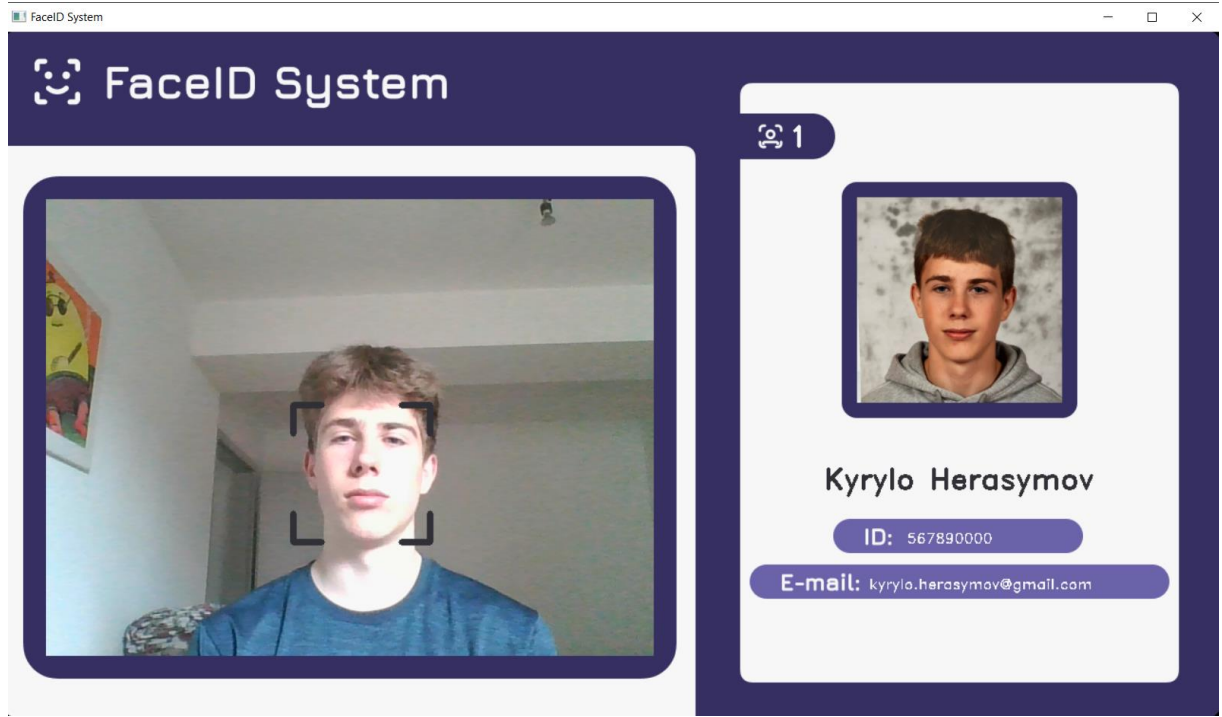


Рисунок 3.37 – Тестування нової зареєстрованої у системі особи

Коли нова особа успішно проходить процедуру входу вперше, її дані автоматично зберігаються до бази даних у реальному часі. Цей процес є важливим для забезпечення ефективної роботи системи безпеки та підтримки актуальності інформації (рис. 3.38).

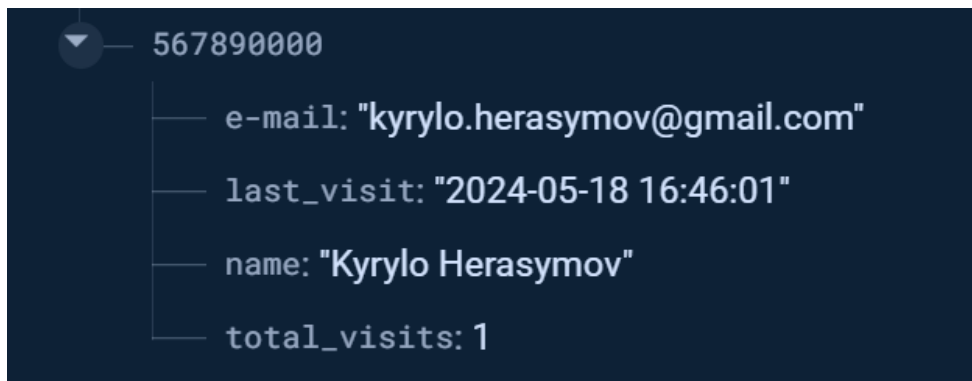


Рисунок 3.38 – Дані нової зареєстрованої у системі особи у БД

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено застосунок для онлайн розпізнавання облич у відеопотоці. Проєкт включав імплементацію алгоритму детекції HOG, ідентифікації ключових точок облич Face Landmark Estimation та метод класифікації SVM з метою забезпечення надійної та ефективної системи виявлення осіб у реальному часі.

Аналіз реалізованих алгоритмів підтвердив їхню високу ефективність і точність у роботі з відеопотоками, а також здатність системи швидко адаптуватися до змін у зовнішніх умовах, таких як освітлення та ракурс зйомки. Використання сучасних бібліотек та мов програмування, таких як OpenCV та Python, сприяло швидкій розробці та гнучкості системи, що дозволило впровадити комплексні функції обробки зображень і машинного навчання без зниження продуктивності.

Тестування програми підтвердило, що розроблена система здатна ефективно функціонувати у різних умовах освітлення та при різних ракурсах камери, забезпечуючи високу точність розпізнавання. Також система виявилася стійкою до шумів на зображенні та інших візуальних спотворень.

Розроблена система може бути використана як основа для подальшого розвитку та інтеграції з іншими технологіями та системами. У подальшому планується розширити функціонал системи, додавши можливості глибшого аналізу емоцій, віку та статі виявлених осіб, що зробить систему ще більш універсальною та затребуваною на ринку. Також важливо продовжити роботу над оптимізацією алгоритмів для подальшого підвищення швидкості обробки даних і зменшення вимог до обчислювальних ресурсів, щоб забезпечити більшу доступність та економічну ефективність рішення.

Результати роботи апробовано у вигляді тез доповідей під час V Міжнародної науково-практичної конференції «SCIENCE AND SOCIETY: MODERN TRENDS IN A CHANGING WORLD» [32].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Путятін, Є.П., Гороховатський, В.О., & Матат, О.О. (2006). Методи та алгоритми комп'ютерного зору: навч. посібник.
2. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ*.
3. Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
4. Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5525-5533).
5. Liu, Z., Luo, P., Wang, X., & Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15(2018), 11.
6. Яковлєв, Д., & Щербаков, О. (2022). Методи та алгоритми вирішення задачі розпізнавання обличчя осіб на зображенні. Матеріали Міжнародної науково-практичної конференції «Інформаційні технології та системи». С. 22.
7. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. URL: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78> (дата звернення 10.04.2024).
8. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників.
9. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.
10. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

11. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
12. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.
13. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
14. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.
15. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
16. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.
17. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.
18. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

19. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.
20. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., and Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.
21. Pang, Y., Yuan, Y., Li, X., & Pan, J. (2011). Efficient HOG human detection. *Signal processing*, 91(4), 773-781.
22. Wang, X., Han, T. X., & Yan, S. (2009, September). An HOG-LBP human detector with partial occlusion handling. In 2009 IEEE 12th international conference on computer vision (pp. 32-39). IEEE.
23. Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1867-1874).
24. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
25. Chandra, M. A., & Bedi, S. S. (2021). Survey on SVM and their application in image classification. *International Journal of Information Technology*, 13(5), 1-11.
26. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for fast metric data search in structural methods for image classification. *IEEE Access*, 10, 124738-124746.
27. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification.
28. Nguyen, Q. (2019). Hands-on application development with PyCharm: Accelerate your Python applications using practical coding techniques in PyCharm. Packt Publishing Ltd.

29. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. «O'Reilly Media, Inc.».

30. CVZone. URL: <https://github.com/cvzone/cvzone> (дата звернення 14.04.2024).

31. Moroney, L., & Moroney, L. (2017). An Introduction to Firebase. The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform, 1-24.

32. Герасимова С. С., & Шафроненко А. Ю. (2024). Розробка застосунку для онлайн розпізнавання облич у відеопотоці. Матеріали V Міжнародної науково-практичної конференції «Science and society: modern trends in a changing world». С. 151.