

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Карнидал Михайло Сергійович

Програмний модуль виявлення облич системою комп'ютерного зору/ Software module for face detection by computer vision system

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-42
Карнидал М. С.

Науковий керівник:
к.т.н. Д.І. Загородня

Дипломний проект допущено до
захисту

«___» _____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«_____» _____ 2023р.

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТЦІ

Карнидалу Михайлу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту: Програмний модуль виявлення облич системою комп'ютерного зору / Software module for face detection by computer vision system

керівник проекту к.т.н., доцент кафедри Д.І. Загородня

затверджені наказом по університету від 08 грудня 2023 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- провести аналіз предметної області;
- провести аналіз методів та засобів виявлення обличчя;
- зробити постановку задач дослідження;
- розробити архітектуру системи;
- розробити алгоритмічне забезпечення системи;
- провести реалізацію програмного забезпечення;
- розробити та реалізувати інтерфейс користувача;
- провести налаштування та запуск програмного модуля;
- провести тестування програмного модуля.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- алгоритм роботи модуля виявлення облич;
- діаграма класів системи виявлення обличчя.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Д.І. Загородня		

7. Дата видачі завдання 08 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз систем комп'ютерного зору	30.12.2022	
2	Алгоритмічне та інформаційне забезпечення модуля	24.03.2023	
3	Розробка програмного модуля виявлення облич	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ М.С. Карнидал
(підпис)

Керівник проекту _____ Д.І. Загородня
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 58 с., 38 рис., 3 додатки, 21 джерело.

Метою дипломного проекту є розробка програмного модуля виявлення облич.

Об'єкт дослідження – процеси виявлення облич в автоматизованих системах комп'ютерного зору.

Предмет дослідження – методи та засоби для виявлення облич в системах комп'ютерного зору.

Розроблено та досліджено програмне забезпечення для виявлення облич системою комп'ютерного зору.

Запропонована система надає користувачеві можливість виявлення облич системою комп'ютерного зору.

ВИЯВЛЕННЯ ОБЛИЧ, КОМП'ЮТЕРНИЙ ЗІР, СИСТЕМА, ПРОГРАМНИЙ МОДУЛЬ, ШТУЧНИЙ ІНТЕЛЕКТ.

ABSTRACT

Explanatory note to the diploma project: 58 pp., 38 figures, 3 appendices, 21 sources.

The goal of the diploma project is the development of a face detection software module.

The object of research is face detection processes in automated computer vision systems.

The subject of research is methods and tools for detecting faces in computer vision systems.

Developed and researched software for face detection by computer vision system.

The proposed system provides the user with the ability to detect faces with a computer vision system.

FACE DETECTION, COMPUTER VISION, SYSTEM, SOFTWARE MODULE, ARTIFICIAL INTELLIGENCE.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Програмний модуль виявлення облич системою комп'ютерного зору.

1.2 Область застосування – комп'ютерний зір.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка програмного модуля виявлення облич системою комп'ютерного зору.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

- зручність та зрозумілість представлених результатів для користувача;
- можливість передачі даних та отриманих результатів роботи програмного модуля у файл – можливість експорту даних;
- здатність програмного модуля до подальших змін та доповнень в майбутньому, тобто дотримання принципу модульності;

- здатність програми інтегруватись в систему комп'ютерного зору;
- вхідна інформація отримується шляхом завантаження зображення із бази або відеокамери.

5.2 Вимоги до апаратних засобів:

- модуль повинен працювати на IBM-сумісних робочих станціях;
- мінімальні вимоги до робочих станцій: процесор від 2 ГГц, оперативна пам'ять від 1 Гб, відеокарта від 512 Мб, об'єм пам'яті на жорсткому диску до 1 Гб, клавіатура, маніпулятор «миша», відеокамера.

5.3 Вимоги до програмних засобів:

- для розробки програмне забезпечення - Python 3.11;
- для створення графічного інтерфейсу користувача використано – tkinter;
- для реалізації виявлення обличчя – бібліотеку OpenCV.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ М.С. Карнидал
(підпис) (прізвище та ініціали)

Керівник дипломного проекту _____ Д.І. Загородня
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 АНАЛІЗ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ	11
1.1 Коротка характеристика систем з функцією виявлення облич.....	11
1.2 Опис предметної області.....	12
1.3 Огляд і аналіз існуючих аналогів.....	15
1.4 Постановка задачі дослідження.....	22
2. АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ.....	25
2.1 Загальна структура проектованого модуля.....	25
2.2 Алгоритмічне забезпечення модуля	28
2.3 Інформаційне забезпечення модуля.....	39
3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ВИЯВЛЕННЯ ОБЛИЧ	41
3.1 Реалізація програмного забезпечення.....	41
3.2 Інтерфейс користувача	42
3.3 Тестування програмного забезпечення	44
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
Додаток А Алгоритм роботи модуля виявлення облич	55
Додаток Б Діаграма класів системи виявлення облич.....	56
Додаток В Код модуля.....	57

					ДП.КН.9500032.058.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Карнидал М.С.			Програмний модуль виявлення облич системою комп'ютерного зору	Літ.	Аркуш	Аркушів
Перевір.		Загородня Д.І.					8	58
Консульт.						ЗУНУ.ФКІТ.КН-42		
Н. Контр.		Загородня Д.І.						
Затверд.		Комар М.П.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API - прикладний програмний інтерфейс

CNN - метод згорткових нейронних мереж

LBP – локальні бінарні патерни

ПЗ – програмне забезпечення

ЕОМ – електронно обчислювальна машина

					ДП.КН.9500032.058.ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Виявлення облич стало одним із обов'язкових завдань систем комп'ютерного зору. В першу чергу це стосується забезпечення безпеки в різних сферах, включаючи публічну безпеку, контроль доступу та ідентифікацію осіб. Модулі виявлення облич використовуються в соціальних мережах та фотографічних додатках для автоматичного розміщення та маркування облич на фотографіях. Широке застосування таких модулів спостерігається в рекламі та персоналізованих послугах, коли здійснюється аналіз обличчя користувачів для надання відповідної реклами або пропозиції товарів що відповідають їхнім інтересам. Також такі модулі широко використовується в ігровій індустрії та технологіях доповненої реальності для розпізнавання облич гравців та інтегрування їх у віртуальний світ. В медичній сфері, для розпізнавання певних характеристик або ознак, пов'язаних зі станом здоров'я пацієнта.

Метою дипломного проекту є розробка програмного модуля виявлення облич.

Завдання дипломної роботи полягає у аналізі предметної області, аналізі методів та засобів виявлення облич, розробці архітектури системи, розробці алгоритмічного забезпечення системи, реалізації програмного забезпечення, а також налаштуванні та запуск програмного модуля та його тестуванні.

Об'єктом дослідження є система комп'ютерного зору що має функцію виявлення облич.

Предметом дослідження є програмний модуль, який входить до складу системи комп'ютерного зору і відповідає за виявлення облич.

					ДП.КН.9500032.058.ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ

1.1 Коротка характеристика систем з функцією виявлення облич

В умовах глобалізації та розвитку науково-технічного прогресу технології штучного інтелекту, зокрема системи комп'ютерного зору, набирають все більшу популярність та отримують все більші можливості впровадження у різних сферах. Одним із основних елементів таких систем є процес виявлення облич [1].

Виявлення облич є одним з ключових компонентів систем комп'ютерного зору, які спеціалізуються на впізнаванні та локалізації людей на зображеннях або відео. Основна мета полягає у виявленні різних частин обличчя, таких як очі, ніс, рот, а також визначенні їхньої просторової позиції та орієнтації. Сам процес виявлення облич складається з різних алгоритмів та методів, які можуть включати в себе:

- детектори облич, такі як каскадні класифікатори, які шукають характеристичні ознаки облич, такі як контури, кольорові зони або текстурні шаблони.

- методи сегментації, що дозволяють виділити області облич на зображенні шляхом розбиття на сегменти або застосування методів піксельної класифікації.

- методи вирівнювання облич, що коригують розташування облич для забезпечення однакового масштабу, орієнтації та позиції для подальшого аналізу.

- класифікатори або моделі глибокого навчання, які використовуються для ідентифікації облич із використанням навчальних даних.

Після виявлення облич, результати можуть використовуватися для подальшого аналізу, такого як розпізнавання осіб, виявлення емоцій, визначення руху та багато іншого. Виявлення облич використовується в багатьох сферах, включаючи системи безпеки, контроль доступу, комп'ютерну візуалізацію, розваги та багато інших додатків.

					ДП.КН.9500032.058.ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Сама галузь комп'ютерного зору відносно молода та різноманітна. Існує безліч методів для різних задач комп'ютерного зору, і вони постійно модифікуються і доповнюються. Серед практичних застосувань систем комп'ютерного зору виділяють автоматичне планування і прийняття рішень в системах, що включають механічні дії, такі як рух роботів, безпілотні пристрої, конвеєри та багато інших [1].

1.2 Опис предметної області

Основні компоненти системи комп'ютерного зору з функцією виявлення обличч представлені на рисунку 1.1.

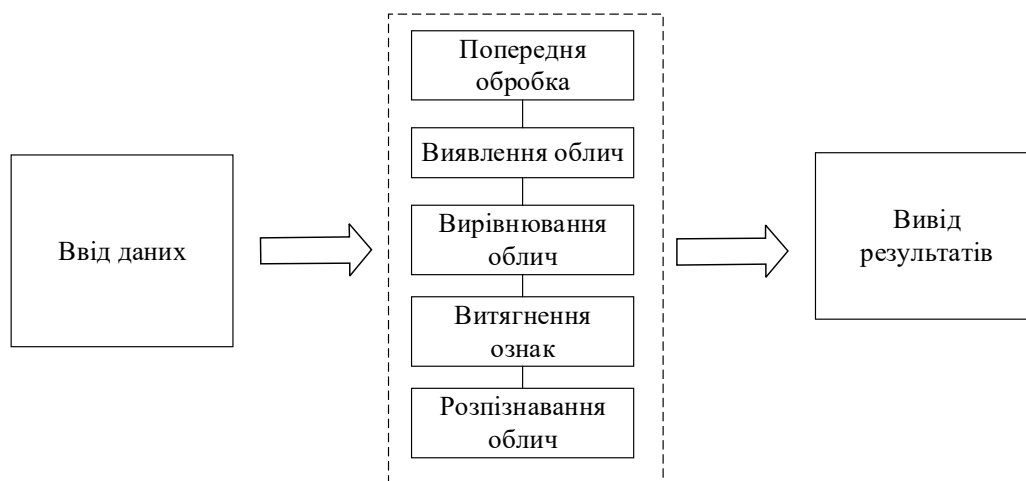


Рисунок 1.1 – Компоненти системи комп'ютерного зору з функцією виявлення обличч

Для збору інформації можуть використовуватись відео та фото камери, вхідні потоки відеоданих. Тому система працює або із зображенням або з відео.

Наступним компонентом є власне робота з вхідними даними. Вона включає: 1). попередню обробку даних для покращення якості зображення або видалення шуму (фільтрація, нормалізація або інші методи покращення якості зображення).

Система ідентифікації обличчя — це технологія, яка дозволяє ідентифікувати обличчя людини з цифрового зображення або відеокадру з джерела відео.

Каскадний класифікатор або каскад посилених класифікаторів, що працюють із властивостями, подібними до Хаара, є окремим випадком ансамблевого навчання, яке називається бустуванням. Зазвичай він покладається на класифікатори Adaboost (та інші моделі, такі як Real Adaboost, Gentle Adaboost або Logitboost). Каскадні класифікатори навчаються на кількох сотнях зразків зображень, які містять об'єкт, який ми хочемо виявити, та інших зображень, які не містять цих зображень.

Для того, щоб визначити чи є обличчя чи ні, існує алгоритм, який називається структурою виявлення об'єктів Віоли–Джонса, який включає всі кроки, необхідні для виявлення живих облич (рисунок 1.3):

- вибір функцій Хаара або функції, отримані з вейвлетів Хаара;
- створення цілісного образу;
- навчання Adaboost;
- каскадні класифікатори.

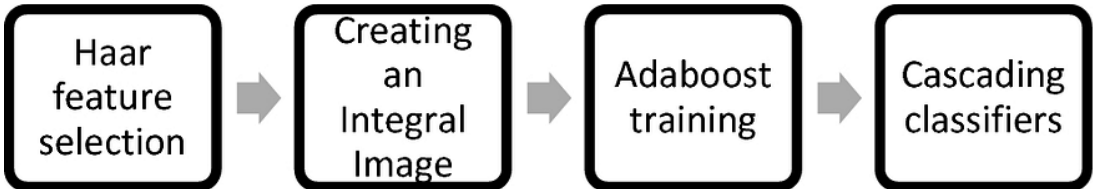


Рисунок 1.3 – Структура методу виявлення об'єктів Віоли–Джонса

Класифікатор Haar Cascade базується на техніці Haar Wavelet для аналізу пікселів у зображенні на квадрати за функціями. Це використовує концепції «цілісного зображення» для обчислення виявлених «функцій».

Каскадний класифікатор Хаара базується на алгоритмі виявлення Віоли–Джонса, який навчається на певних введених обличчях і необличчях і навчає класифікатор, який ідентифікує обличчя. Замість того, щоб застосовувати всі 6000 функцій у вікні, функції групуються в різні етапи класифікаторів і

									ДП.КН.9500032.058.ПЗ	Арк.
										14
Зм.	Арк.	№ докум.	Підпис	Дата						

застосовуються один за одним (рисунок 1.4). Зазвичай перші кілька етапів містять дуже малу кількість функцій. Якщо вікно не проходить перший етап, то його відхиляють та не розглядають інші функції на ньому. Якщо він проходить, то застосовується другий етап функцій і цей процес продовжується. Вікно, яке проходить усі етапи і є областю обличчя.

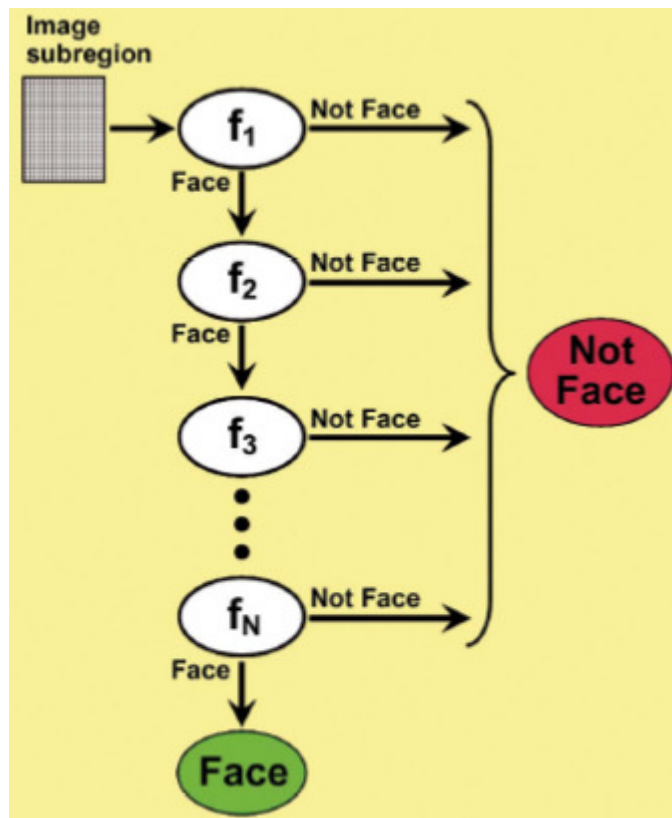


Рисунок 1.4 – Каскадний класифікатор Хаара

1.3 Огляд і аналіз існуючих аналогів

Amazon Rekognition[2] - це хмарне програмне забезпечення - сервісна платформа комп'ютерного зору, яка була запущена в 2016 році. На рисунку 1.5 зображено інтерфейс користувача Amazon Rekognition.

						ДП.КН.9500032.058.ПЗ	Арк.
							15
Зм.	Арк.	№ докум.	Підпис	Дата			

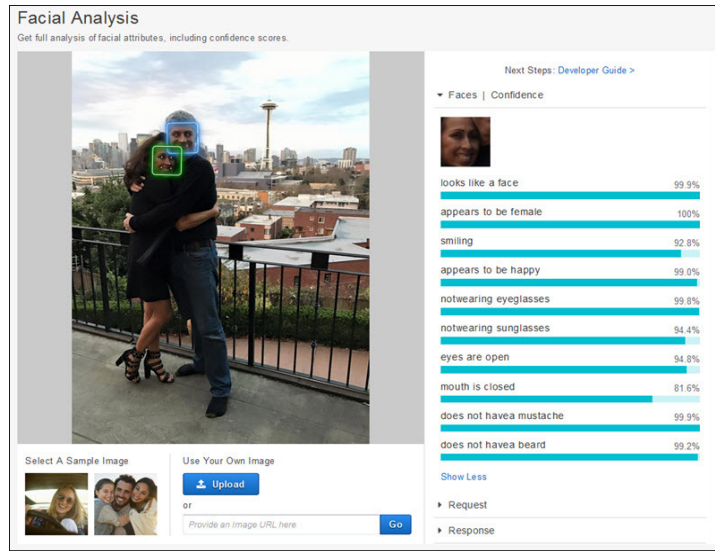


Рисунок 1.5 – Інтерфейс Amazon Rekognition

Його було продано та використовується низкою урядових установ Сполучених Штатів, зокрема Імміграційною та митною службою США та поліцією Орландо, штат Флорида, як а також приватні особи. Rekognition надає ряд можливостей комп'ютерного зору, які можна розділити на дві категорії: алгоритми, попередньо навчені на даних, зібраних Amazon або її партнерами, і алгоритми, які користувач може навчити на спеціальному наборі даних. Деякі з основних функцій Amazon Rekognition:

- розпізнавання облич: система може розпізнавати обличчя на фотографіях та відео, а також виконувати порівняння обличчя для ідентифікації користувачів. Вона також може розпізнавати емоції на обличчях та визначати інші параметри, такі як вік та стать.
- розпізнавання об'єктів: система може розпізнавати об'єкти на зображеннях та відео, такі як автомобілі, будівлі, тварини та інші.
- аналіз зображень: система може аналізувати зображення та визначати різноманітні параметри, такі як кольори, форми та текстури.
- виділення облич та об'єктів: система може виділяти та вирізувати обличчя та об'єкти на зображеннях та відео.
- аналіз відео: система може аналізувати відео та визначати параметри, такі як кількість людей на кадрі, їх рух та інші характеристики.

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

Amazon Rekognition може бути використана в різних галузях, наприклад:

- В системах безпеки: система може використовуватись для ідентифікації осіб на вході до будівлі або на території об'єкта.
- В медичній сфері: система може використовуватись для розпізнавання хвороб на зображеннях медичних досліджень.
- В транспорті: система може використовуватись для аналізу трафіку на дорогах, зокрема, визначення кількості автомобілів, їх швидкості та руху.
- В електронній комерції: система може використовуватись для підвищення рівня персоналізації рекламних кампаній, зокрема, для аналізу зображень продуктів, які користувачі переглядають або купують.
- В мультимедійній індустрії: система може використовуватись для автоматичного розмічення фото та відео, зокрема, для ідентифікації обличь, місць та об'єктів на зображеннях.

Amazon Rekognition має ряд переваг, зокрема:

- Швидкість: система може обробляти велику кількість зображень та відео швидко та ефективно завдяки використанню технологій глибинного навчання та машинного навчання.
- Надійність: система має високу точність розпізнавання обличь та об'єктів, що дозволяє використовувати її в критичних ситуаціях, наприклад, в системах безпеки.
- Масштабованість: система може масштабуватись в залежності від потреб користувача та обсягу даних, що оброблюються.
- Інтеграція з іншими сервісами Amazon: Amazon Rekognition може легко інтегруватись з іншими сервісами Amazon, такими як Amazon S3, Amazon Lambda та інші.

Проте варто зазначити, що Amazon Rekognition також має деякі недоліки, зокрема, можливість помилкового розпізнавання обличь та об'єктів, а також проблеми з конфіденційністю даних, оскільки система може зберігати та обробляти особисту інформацію користувачів.

					ДП.КН.9500032.058.ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

Ціна на Amazon Rekognition залежить від обсягу даних, що оброблюються та типу оплати. Amazon пропонує два варіанти оплати: за використання та за об'єм. За використання передбачає оплату за кожну хвилину використання сервісу. Ціна становить \$0,001 за хвилину на використання обробки зображень та \$0,0012 за хвилину на використання обробки відео. Оплата за об'єм передбачає оплату за кожен оброблений об'єкт (зображення чи відео). Ціна становить \$0,001 на зображення та \$0,10 на хвилину відео. Усі операції з Amazon Rekognition виконуються через API, тому для користування сервісом необхідно мати досвід у програмуванні та розробці програмного забезпечення. Незважаючи на деякі недоліки та певні ризики, пов'язані з конфіденційністю даних, Amazon Rekognition є потужним та ефективним сервісом для розпізнавання облич та об'єктів, що знайшов широке застосування в різних сферах.

FaceNet [3] - це система розпізнавання облич, яка використовує нейронні мережі для створення векторних представлень облич, що дозволяє більш точно ідентифікувати обличчя, на рисунку 1.6 зображено інтерфейс користувача FaceNet.

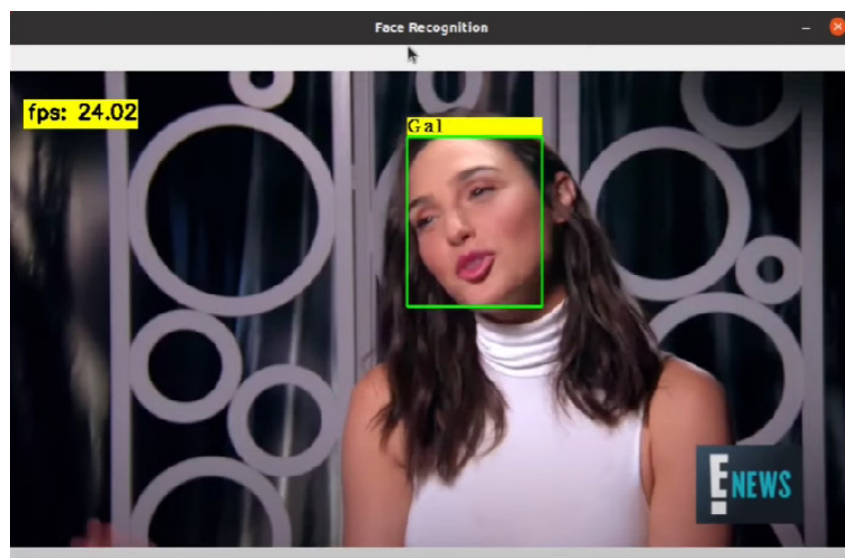


Рисунок 1.6 – Інтерфейс користувача FaceNet

FaceNet може працювати з будь-якими зображеннями облич та навіть з нецілісними зображеннями, де обличчя можуть бути частково приховані або

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

розташовані відносно інших об'єктів на зображенні. Система може працювати з великим обсягом даних та використовувати їх для тренування, що дозволяє досягти високої точності розпізнавання обличь. Векторне представлення обличь дозволяє використовувати алгоритми кластеризації та ранжування для подальшої обробки даних. FaceNet також має можливість використовувати не тільки зображення обличь, але й інші візуальні дані, такі як відео та 3D-моделі. FaceNet використовує вже наявні нейронні мережі, такі як Inception та ResNet, для виявлення обличь та екстракції їх векторних представлень. Далі використовується техніка, що називається Triplet Loss, яка допомагає зменшити відстань між векторними представленнями обличчя однієї людини та збільшити відстань між векторними представленнями обличь різних людей. Це дозволяє системі розрізняти обличчя та розпізнавати їх. FaceNet знаходить застосування в різних сферах, включаючи безпеку, відеоспостереження, рекламу та діагностику медичних захворювань. Ця система може бути використана для розпізнавання облич у режимі реального часу та для аналізу великих обсягів даних, що потребують обробки. FaceNet також використовується для створення систем ідентифікації осіб та розпізнавання емоцій на зображеннях облич. Зокрема, він може використовуватися для автоматичної ідентифікації людей на фотографіях та відео з метою встановлення їхньої особистості. Наприклад, у сфері безпеки FaceNet може використовуватися для автоматичного виявлення злочинців на відеозаписах з камер відеоспостереження. У сфері медицини FaceNet може використовуватися для діагностики захворювань шляхом аналізу фотографій облич пацієнтів. Незважаючи на багато переваг, FaceNet також має свої обмеження. Зокрема, він може працювати не так добре з даними, що містяться в гучних та поганих освітленнях. Також він може працювати не так добре з низькоякісними зображеннями або з зображеннями, на яких обличчя приховані частково або повністю. У цілому, FaceNet - це потужна система розпізнавання облич, яка має великий потенціал в різних сферах застосування, включаючи безпеку, медицину та рекламу.

					ДП.КН.9500032.058.ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

Kairos [4] - це ще одна система розпізнавання обличь, яка використовується в комерційних цілях. Вона використовує машинне навчання та нейронні мережі для розпізнавання обличь, на рисунку 1.7 зображено інтерфейс користувача системи розпізнавання обличь Kairos.



Рисунок 1.7 – Інтерфейс користувача Kairos

Система Kairos має такі функції:

- Розпізнавання обличчя та ідентифікація: система може визначити, чи присутній на зображенні обличчя, та зіставити його зі збереженими зображеннями обличчя для ідентифікації користувача.
- Емоційний аналіз: система може розпізнавати емоції на обличчі та визначати їх інтенсивність.
- Аналіз параметрів обличь: система може визначати такі параметри, як вік, стать, етнічна приналежність тощо.

Kairos може бути використана в різних галузях, наприклад:

- В системах безпеки: система може використовуватись для ідентифікації користувачів на вході до будівлі або на території об'єкта.
- В банківських системах: система може використовуватись для ідентифікації клієнтів банку та підтвердження їх особи.

						ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			20

- В маркетингу та рекламі: система може використовуватись для збору даних про аудиторію та аналізу їхньої поведінки.

Однак, варто зазначити, що системи розпізнавання обличчя можуть мати певні проблеми з приватністю та безпекою даних, тому використання їх повинно дотримуватись відповідних правил та норм.

Microsoft Face API [5] - це хмарна служба, яка використовує машинне навчання для розпізнавання обличчя та інших аналітичних завдань, на рисунку 1.8 зображено інтерфейс користувача Microsoft Face API.

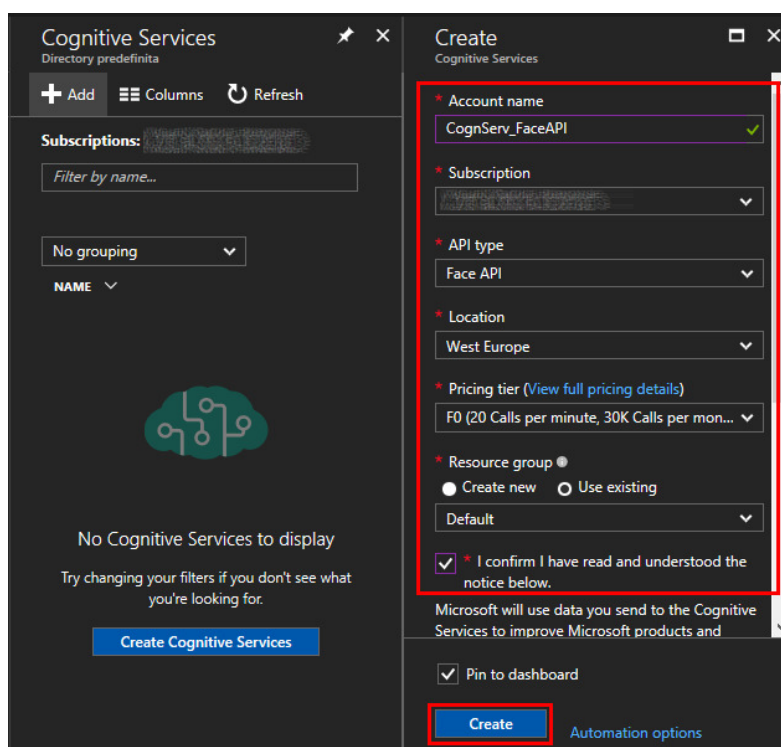


Рисунок 1.8 – Інтерфейс Microsoft Face API

Microsoft Face API - це хмарна система розпізнавання обличчя, яка розроблена Microsoft. Вона дозволяє розпізнавати та аналізувати обличчя на фотографіях та відео, а також визначати їхні риси, такі як вік, стать, емоції, пози та інші. Для роботи з Microsoft Face API потрібно мати ключ доступу, який можна отримати на офіційному сайті Microsoft. Після цього ви можете використовувати API, викликаючи його за допомогою REST-запитів з вашого додатку. Основні можливості Microsoft Face API:

						ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			21

- Розпізнавання облич: система може розпізнавати обличчя на фотографіях та відео, а також визначати їхні координати.
- Визначення рис облич: API може визначати такі риси, як вік, стать, емоції, пози, форма обличчя та інші.
- Ідентифікація осіб: система може ідентифікувати конкретних осіб на фотографіях та відео, якщо вони вже були додані до бази даних.
- Аналіз емоцій: система може аналізувати емоції на обличчі, такі як радість, сум, злість та інші.
- Виявлення обличчя: API може виявляти обличчя на фотографіях та відео навіть у складних умовах, таких як погане освітлення або зміщеність.

Microsoft Face API може бути використаний в різних галузях, включаючи безпеку, медицину, рекламу та інші. Оскільки він є частиною екосистеми Microsoft Azure, ви можете використовувати його разом з іншими хмарними сервісами, такими як Azure Machine Learning та Cognitive Services, для розширення можливостей своєї програми.

1.4 Постановка задачі дослідження

Теоретична постановка задачі полягає у визначенні теоретичної бази, необхідної для розробки програмного модуля виявлення обличчя системою комп'ютерного зору. У цьому розділі розглянуто поняття комп'ютерного зору, методи та алгоритми виявлення обличчя, основні принципи роботи системи комп'ютерного зору, що допоможе читачу краще зрозуміти технології, що використовуються в дипломній роботі.

Практична постановка задачі має на меті описати процес розробки програмного модуля для виявлення обличчя системою комп'ютерного зору. У цьому розділі слід детально описати мету і завдання дослідження, вибір технологій та інструментів, які були використані для розробки програмного модуля, та процес його розробки. Також, важливим елементом цього розділу є

					ДП.КН.9500032.058.ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

опис тестування та оцінка результатів роботи програмного модуля, де слід порівняти його результати з результатами інших аналогічних рішень. Шляхи вирішення поставленої задачі можуть включати наступні етапи:

- Збір та обробка даних - необхідно зібрати велику кількість фотографій з обличчями, підготувати їх для подальшої обробки (наприклад, обрізати, змінити формат, підвищити роздільну здатність).

- Вибір методів та алгоритмів виявлення обличь - для вирішення поставленої задачі можна використовувати різноманітні алгоритми виявлення обличь.

- Розробка програмного модуля - на основі вибраних методів та алгоритмів слід розробити програмний модуль виявлення облич системою комп'ютерного зору. У цьому процесі можуть використовуватись різноманітні інструменти, такі як OpenCV.

- Тестування та оцінка результатів - після розробки програмного модуля необхідно провести його тестування з використанням підготовлених даних. Результати тестування слід порівняти з результатами інших аналогічних рішень, а також з вимогами, які були поставлені в постановці задачі.

- Аналіз результатів та вдосконалення програмного модуля - на основі результатів тестування можна виявити помилки та недоліки програмного модуля, а також знайти шляхи його вдосконалення. У цьому етапі можуть бути внесені зміни до програмного коду, виконані додаткові тести та оцінка результатів.

- Визначення параметрів програмного модуля - для досягнення максимальної ефективності програмного модуля, необхідно визначити оптимальні параметри. Це може включати в себе вибір розміру вікна, встановлення порогових значень для виявлення обличь, встановлення швидкості роботи та інші параметри.

- Розробка інтерфейсу користувача - для зручності користувачів можна розробити інтерфейс користувача програмного модуля. Це може бути простий

					ДП.КН.9500032.058.ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

інтерфейс для запуску програми та вибору параметрів, або більш складний з можливістю відображення результатів в режимі реального часу.

- Аналіз застосування - для додаткового дослідження можна провести аналіз застосування розробленого програмного модуля у різних сферах, таких як безпека, медицина, реклама, наука та інші.

					ДП.КН.9500032.058.ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

полягає у поліпшенні зображення та зменшенні фонового шуму, щоб покращити точність виявлення облич;

– виявлення облич – основний етап модуля виявлення облич, включає використання відповідних алгоритмів та моделей для виявлення облич на зображенні або відеопотоці. Тут можуть використовуватися різні методи, такі як haarcascades, нейронні мережі (наприклад, Convolutional Neural Networks - CNN), методи засновані на ознаках тощо;

– помітки та рамки навколо облич. Після виявлення облич, модуль може виконати додаткову обробку, яка включає позначення облич на зображенні за допомогою поміток або рамок. Це дозволяє виділити та показати місцезнаходження облич на вихідному зображенні.

– Відображення результатів – останній крок, який полягає у відображенні результатів виявлення облич. Це може включати відображення зображення з позначеними обличчями або передачу результатів до наступних етапів обробки або аналізу.

Модуль виявлення облич може бути частиною більшої системи комп'ютерного зору і інтегруватися з іншими модулями, такими як модуль розпізнавання облич або модуль аналізу облич. Це дозволяє використовувати результати виявлення облич в подальшому аналізі, ідентифікації або взаємодії з користувачем.

Між функціями програмного модуля виявлення облич системою комп'ютерного зору передаються різні дані для виконання обробки та обміну інформацією. Основні дані, які можуть передаватися між функціями, включають:

– зображення або відеопотік. Це основний вхідний параметр, який передається з функції завантаження до функції попередньої обробки. Зображення може бути представлене у вигляді масиву пікселів або спеціального об'єкта, що містить інформацію про зображення. Відеопотік може бути представлений як послідовність кадрів.

– оброблене зображення. Цей параметр представляє попередньо оброблене зображення та передається з функції попередньої обробки до функції

					ДП.КН.9500032.058.ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

виявлення облич. Оброблене зображення може мати змінений розмір, кольоровий простір або будь-які інші зміни, залежно від попередньої обробки.

– виявлені обличчя. Це результат функції виявлення облич і представляється у вигляді масиву або списку з обличчями, які були виявлені на зображенні. Кожне обличчя може бути представлене у вигляді координат або рамки, які вказують його положення на зображенні.

– позначене зображення. Це параметр, який передається з функції виявлення облич до функції відображення результатів. Воно містить зображення з позначеними обличчями або рамками, що вказують їх положення.

Ці дані передаються між функціями для обробки, аналізу та відображення результатів виявлення облич. Крім цього, додаткові параметри можуть бути передані для керування параметрами виявлення, позначення або відображення облич. Наприклад, це може включати розмір рамки, порогові значення для виявлення, параметри фільтрації тощо.

Data Flow Diagram (DFD) програмного модуля містить послідовність обробки даних у програмному модулі виявлення облич. Вхідне зображення передається на вхід модулю, яке виконує попередню обробку зображення. Після цього виконується функція виявлення облич, яка визначає положення облич на обробленому зображенні.

Потім результати виявлення передаються до модулю позначення облич, який може ставити помітки або розміщувати рамки навколо облич. Нарешті, дані з позначеними обличчями передаються до модулю відображення результатів, де вони можуть бути показані на екрані або збережені у вихідному файлі зображення.

Діаграма переходів станів (State Transition Diagram) програмного модуля виявлення облич системою комп'ютерного зорумістить різні стани, у яких може перебувати програмний модуль виявлення облич системою комп'ютерного зору. Початковий стан є початковим пунктом, з якого може розпочинатися процес. З нього переходиться до стану "Завантаження зображення", де виконується завантаження вхідного зображення.

					ДП.КН.9500032.058.ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

Після завантаження зображення переходиться до стану "Попередня обробка", де зображення попередньо обробляється для покращення якості та зменшення шуму. Потім модуль переходить до стану "Виявлення облич", де виконується сам процес виявлення облич на обробленому зображенні.

Якщо обличчя були виявлені, модуль переходить до стану "Позначення облич", де обличчя позначаються на зображенні, наприклад, розміщуються рамки навколо облич. Нарешті, модуль переходить до стану "Завершений", де результати відображаються або зберігаються, що позначає закінчення процесу виявлення облич.

2.2 Алгоритмічне забезпечення модуля

Алгоритм виявлення облич Віоли-Джонса [6] - це система машинного навчання виявлення об'єктів, запропонована 2001 року Полом Віолою та Майклом Джонсом. Її було вмотивовано головно задачею виявлення облич, хоча її можливо пристосовувати й для виявлення інших класів об'єктів.

Хоч метод Віоли-Джонса і має нижчу точність за сучасніші методи, такі як згортова нейронна мережа, його ефективність і компактний розмір означають, що його все ще використовують у випадках з обмеженою обчислювальною потужністю. Виділяють основні принципи, на яких основана робота даного методу:

- інтегральне представлення зображень;
- пошук облич за допомогою ознак Хаара;
- каскадна класифікація з застосуванням бустингу.

Для обчислення яскравості прямокутної області зображення використовується інтегральне представлення. Інтегральне представлення дозволяє швидко розрахувати загальну яскравість довільного прямокутника на заданому зображенні, при тому що час розрахунку не залежить від площі прямокутника. Інтегральне представлення зображення являє собою матрицю,

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

розміри якої співпадають з розмірами вхідного зображення. У кожному елементі такої матриці зберігається сума інтенсивностей пікселів, які знаходяться лівіше і вище даного елемента. Елементи матриці розраховуються у відповідності з формулою:

$$I(x, y) = \sum_{x \leq (x', y') \leq y} i(x', y')$$

де $I(x, y)$ – значення точки (x, y) інтегрального зображення; $i(x, y)$ – значення інтенсивності вхідного зображення.

Використання інтегрального представлення зображення дозволяє обчислювати ознаки однакового виду, але з різними геометричними параметрами, так як розрахунок матриці інтегрального представлення займає лінійний час, пропорціональне числу пікселів у зображенні.

Ознаки Хаара - ознаки цифрового зображення, які використовуються для розпізнавання образів. Своєю назвою вони завдячують інтуїтивною подібністю до вейвлетів Хаара. Ознакою f об'єкта a називають відображення $f: A \rightarrow D_f$, де D_f - множина допустимих значень ознаки. Якщо заданий набір векторів f_1, \dots, f_n , тоді вектор $x = (f_1(a), \dots, f_n(a))$ називається ознаковим описом об'єкта $a \in A$. Вперше використання для виявлення об'єктів ознак заснованих на вейвлетах Хаара, було запропоноване в роботі Папагоргиу в 1998 році. Віола і Джонс адаптували цю ідею в своїй роботі і отримали прямокутні ознаки, названі ознаками Хаара. Зовнішній вид ознак можна побачити на рисунку 2.4.

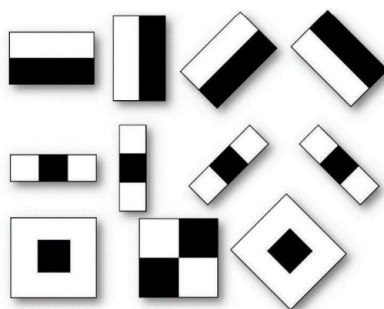


Рисунок 2.4 – Приклад ознак Хаара

У розширеному методі Віоли-Джонса, представленому в бібліотеці комп'ютерного зору OpenCV, використовується також додаткові ознаки, представлені на рисунку 2.5.

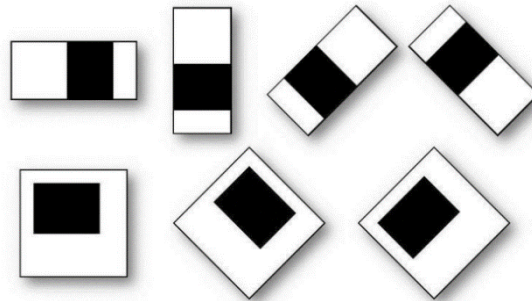


Рисунок 2.5 - Додаткові ознаки Хаари

Результатом обчислення такою ознакою на інтегральному представленні буде:

$$F = U - V,$$

де U – сума значень яскравостей точок, закритий світлою частиною ознаки, а V сума значень яскравостей точок, закритих темною частиною ознаки.

Такі ознаки описують зміну яскравості по двом осям зображення. Пошук обличчя виконується за допомогою так званого скануючого вікна, розміри якого в оригінальному алгоритмі складають 24×24 пікселя. Вікно переміщається по зображенню з кроком в 1 піксель і для кожного його положення обчислюються ознаки Хаара з різним масштабом і положенням у вікні. При цьому сканування проводиться так само і для різних масштабів скануючого вікна. Знайдені ознаки передаються класифікатору, який визначає по їх значенням, чи являється область зображення, яка відноситься до вікна, обличчям чи ні.

Каскадна[7] структура класифікатора дозволяє пришвидшити виявлення обличчя, фокусуючи роботу на найбільш цікавих областях зображення. Каскад надає собою структурну організацію слабких класифікаторів, навчених з застосуванням процедури бустингу. Так при малих обчислювальних витратах

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

можна на ранніх етапах розпізнавання відкинути зображення, з великою долею імовірності, які не містять шуканий об'єкт. Приклад каскадною структури класифікаторів представлений на рисунку 2.6.

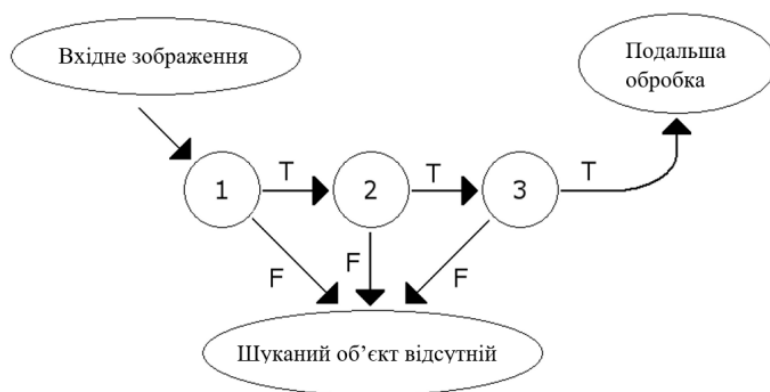


Рисунок 2.6 - Приклад каскадної структури класифікаторів.

Кожний рівень каскаду навчається за допомогою алгоритму AdaBoost. Кількість особливостей, використовуваних в ньому, збільшуватись до тих пір, поки виявлення цільового об'єкта і помилки першого роду не досягнуть заданого значення. Рівні визначається за допомогою тестування поточного детектора на множині, що перевіряється. Якщо загальна помилка першого роду для всього об'єкта ще не досягнута, то в каскад додається ще один шар. Від'ємна множина для навчання наступних шарів визначається шляхом збору всіх хибних виявлень при використанні поточного каскаду. Результатом класифікації буде отриманий набір областей зображення, якій містять об'єкт, який шукається. Далі виключаються вкладені повторення при виявленні одного і того ж об'єкта, які викликані масштабуванням скануючого вікна. Для подальшої обробки знайдені обличчя переводяться в градації сірого і масштабуються до розміру 128*128 пікселів.

Для усунення шумів на зображеннях[8] використовується фільтр Гауса. Фільтр Гауса - це фільтр розмиття зображення, який використовує нормальне розподілення для обчислення перетворення, яке застосовується до кожного

пікселя зображення. Нормальне розподілення для двох вимірів описується формулою:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

де r – радіус розмиття; $r = x^2+y^2$; σ – стандартне відхилення розподілення Гауса.

Ця формула задає поверхню, яка має вигляд концентрованих окружностей з нормальним розподіленням від центральної точки. Пікселі де розподілення відмінне від нуля використовується для побудови матриці згортки, яка застосовується до вхідного зображення. Значення пікселя приймає найбільшу вагу і сусідні пікселі приймають менші ваги, в залежності від відстані до них. Приклад роботи розмиття по Гаусу для одновимірного масиву можна побачити на рисунку 2.7.

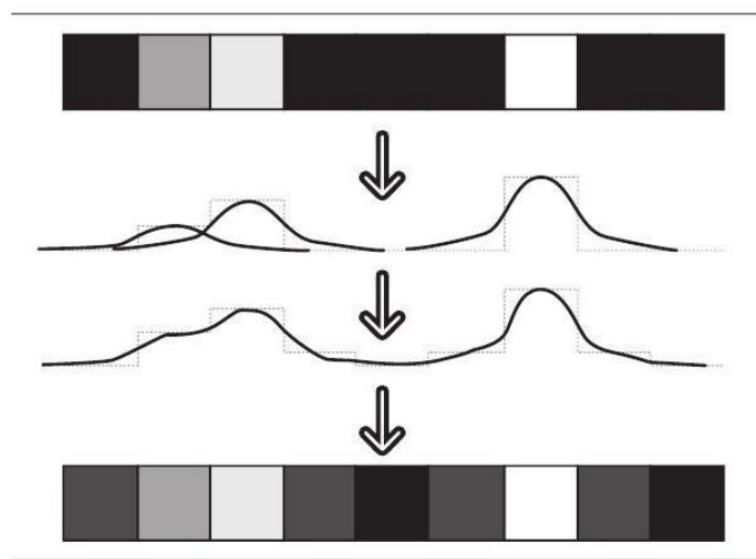


Рисунок 2.7 – Розмиття по Гаусу на однорідному масиві.

Таким чином, розмиття по Гаусу дозволяє позбутися від не бажаних шумів на зображеннях, що зведе до мінімуму їх вплив при подальшій класифікації. Результат застосування фільтра Гауса до цілого зображення є на рисунку 2.8.



Рисунок 2.8 - Результат застосування фільтра Гауса.

LBP[9] оператор вперше був запропонований в 1996 році для класифікації текстур. Суть оператора ґрунтується в застосування до пікселів зображення граничного перетворення, в якому значення яскравості у пікселя, що обробляється, порівнюється зі значенням яскравості пікселів його околиці. Результат порівняння кожного пікселя околиці з пікселем, що обробляється конкатенується у двійкове число. У класичному варіанті використовується квадратна околиця 3*3 пікселя. Приклад розрахунку LBP перетворення для такої околиці наведено на рисунку 2.9.

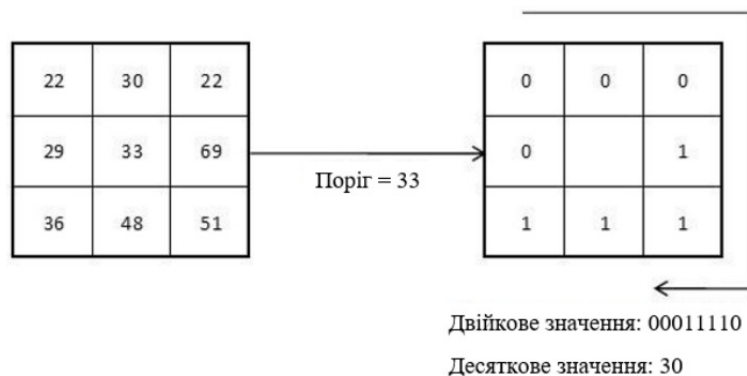


Рисунок 2.9 - Класичний LBP оператор.

Після застосування LBP оператора, зображення ділиться на прямокутні області, які описуються наскільки часто зустрічаються в даній області пікселі різної яскравості. Значення елементів LBP гістограми можуть бути описані наступною формулою:

$$H_j = \sum_{x,y} I\{f(x,y) = i\}, i = 0, \dots, n - 1$$

де $f(x,y)$ – значення яскравості пікселя LBP зображення з координатам (x,y) ; n – кількість різних значень яскравості пікселів; $I\{A\}=1$, якщо A - істина, інакше $I\{A\}=0$.

Отримані дані нормалізуються, конкатенуються і використовуються далі в якості ознак класифікації. Приклад розбиття зображення на прямокутні області і формування гістограм можна побачити на рис. 2.7.

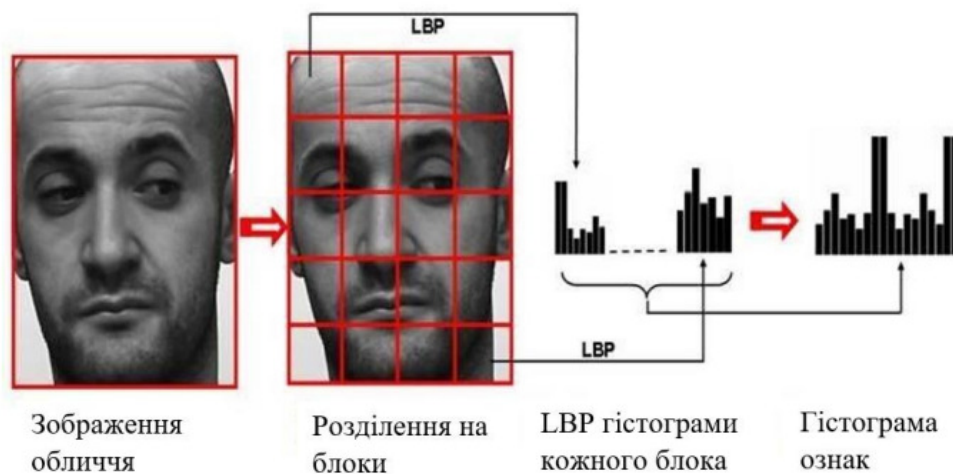


Рисунок 2.10 - Розбиття зображення на прямокутні області формування гістограм.

В результаті виходить опис зображення обличчя на трьох рівнях локалізації. При цьому такий опис не залежить від монотонних змін освітлення.

Суттєву інформацію про форму об'єктів на зображенні несе тільки частина з них. Такі локальні бінарні шаблони були названі рівномірними. До даного виду

LBP відносяться ті шаблони, двійковий код яких містить не більше двох переходів між нулем та одиницею. Вони описуються лише важливі локальні особливості зображення, такі як кінці ліній, грані, плями. Приклади рівномірних LBP представлені на рисунку 2.11.

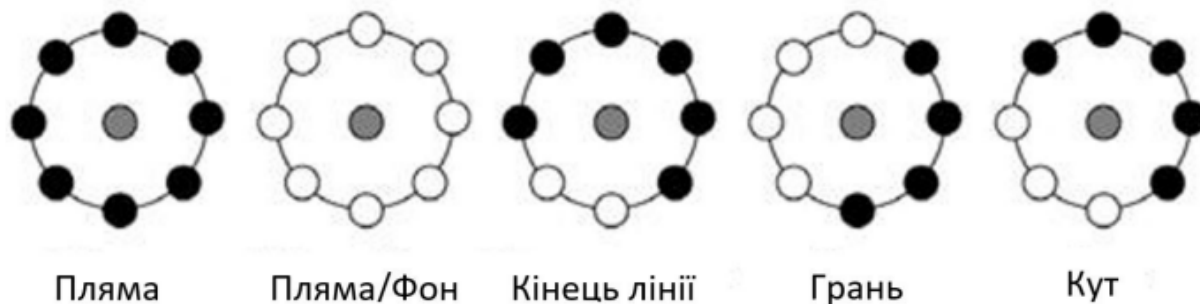


Рисунок 2.11 - Рівномірні LBP.

Всього налічується 58 рівномірних LBP. В результаті є скорочення використання ресурсів ЕОМ ніж в оригінальному алгоритмі. Таке скорочення розмірності дозволяє знизити витрати пам'яті і суттєво збільшити швидкість класифікації, при цьому покращивши її показники за рахунок використання тільки важливих ознак.

Модифікація алгоритму розрахунку локальних бінарних шаблонів дозволяє ще сильні скоротити витрати пам'яті і обчислювальну складність класифікації. Суть модифікації оснований на тому, що в якості граничного значення для кожного пікселя околиці приймається не значення яскравості центрального пікселя околиця, а значення яскравості протилежного відносно центра околиці пікселя. Порівняння розрахунку класичного LBP і центрально-симетричного LBP наведено на рисунку 2.12.

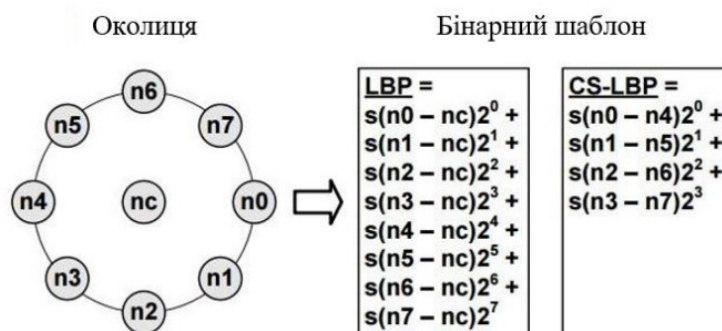


Рисунок 2.12 - Розрахунок значення звичайного і центральносиметричного LBP (CS-LBP)

На рисунку показано, в даному алгоритмі число розрядів значень перетворень пікселів скорочується до чотирьох. Відповідно розмірність гістограми ознак зменшується до 2^4 . В результаті отримується ще більша економія пам'яті і збільшення швидкості класифікації ніж при використанні рівномірних LBP. Однак подібне скорочення розмірності гістограми ознак може негативно вплинути на точність класифікації.

Класичні LBP є оптимальними для використання та мають хороший баланс між точністю та продуктивністю для системи виявлення обличч було обрано саме цю модифікацію.

Зображення облич, які отримуються після процедури виявлення, мають квадратну форму. Однак обличчя займає не весь простір зображення. Тому логічно було б виключити вплив на рішення класифікатора областей зображення, в яких немає обличчя. Для цього потрібно застосувати маски значимих областей зображення [10]. Така маска представляє собою зображення однакового розміру з вхідним зображенням. Пікселі ненульової яскравості в масці відповідає значимим областям. Значимою областю є область в центрі зображення, яка відповідає обличчю. Значення пікселів результуючого зображення можна визначити за формулою:

$$R(x, y) = \begin{cases} I(x, y) & \text{якщо } M(x, y) \neq 0 \\ 0 & \text{якщо } M(x, y) = 0 \end{cases}$$

де $I(x,y)$ – значення яскравості пікселя оброблюваного зображення, $M(x,y)$ - значення яскравості пікселя маски значимих областей.

Маску значимих областей використовується після виконання LBP перетворення. Таким чином, всі незначимі пікселі зображення будуть згруповані в одне значення. Приклад застосування маски значимих областей до зображення, обробленого LBP оператором, наведено на рисунку 2.13.

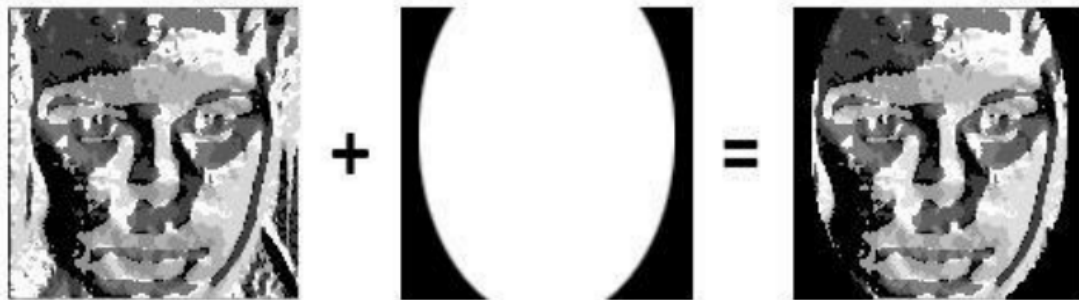


Рисунок 2.13 - Застосування маски значимих областей.

На рисунку можна побачити, що в результаті застосування даної операції пікселі зображення, які не мають впливати на результат класифікації, приймають нульове значення яскравості.

Отримані дані класифікуються методом найближчого сусіда [11]. Метод найближчого сусіда є простим алгоритмом класифікації, суть якого полягає в тому, що об'єкт належить до того класу, до елемента якого він найближче знаходиться. На рисунку 2.14 коло зеленого кольору відповідно до цього алгоритмом повинне бути класифіковане як червоний трикутник.

Метод, в якому об'єкт відносять до класу, до якого відносяться більшість його сусідів в околиці заданого розміру, використовується для покращення результатів. Математично алгоритм можна описати таким чином. На першому кроці визначається елемент x_s навчальної вибірки з N елементів, який найближче до представленого образу x , тобто:

$$\|x - x_s\| = \min\{\|x - x_i\| : i = 1, \dots, N\}$$

							ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата				37

за зображенням 24×24, третя стадія проводить фінальне розпізнавання обличчя/не обличчя, і навіть оцінку описує прямокутника. Блок-схема описаного алгоритму системи виявлення обличчя представлена у додатку А.

2.3 Інформаційне забезпечення модуля

Вхідна інформація, що обробляється в програмному модулі виявлення облич, зображення отримується з камери. Під час передпроцесінгу вхідна інформація обробляється та видаляється від шумів та передається для подальшої обробки [12].

На наступному кроці вхідна «очищена» інформація передається алгоритму для виявлення та відстеження обличчя в кадрі та виводиться інформація на екран.

Вихідна інформація, яка отримується в результаті обробки зображень в програмному модулі виявлення обличчя, включає координати та розміри виявлених облич та їх орієнтацію.

Діаграму класів [14] системи виявлення облич представлено у додатку Б. Як видно з додатку, першим класом системи є клас одержання вхідних даних, наступним класом системи виявлення облич це клас виявлення облич у вигляді представлення вхідних даних двовимірний масив чисел, наступним класом є клас гаусовий фільтр, цей фільтр використовується для мінімізації впливу областей де не виявлено обличчя на подальші результати представлено двовимірному масиві чисел, наступним класом в системі виявлення обличчя є клас LBP transformation у вигляді масиву чисел, маска значимих областей зображення є наступним класом системи виявлення обличчя, яка є представленим в вигляді масиву чисел, наступним класом програмного модулю виявлення обличчя є клас класифікації за допомогою k-середніх представлений у потоці класифікованих даних, завершальним класом даної системи є клас виводу інформації.

					ДП.КН.9500032.058.ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

Для перевірки методів використовують наявні бази зображень, однією таких є база зображень обличь лабораторії Кембриджського Університету. Вона містить зображення 40 осіб, по 10 зображень на кожного. Освітлення на даних зображеннях не змінюється, однак присутні варіації в положенні особи при зйомці.



Рисунок 2.15 - Зображення з бази даних Кембриджського університету

Другий набір даних – база зображень обличь лабораторії Єльського університету. Дана база містить зображення 38 людей, по 65 зображень на кожного, включаючи різні зображення з освітленням. Приклад зображень по 10 зображень на кожну людину з другої бази обличь наведений на рисунку 2.16.



Рисунок 2.16 - Зображень облич лабораторії Єльського університету

3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ВИЯВЛЕННЯ ОБЛИЧ

3.1 Реалізація програмного забезпечення

Для програмної реалізації розробки програмного модулю виявлення обличчя системою комп'ютерного зору було розроблено систему комп'ютерного зору, структурна схема якого зображена на рисунку 3.1

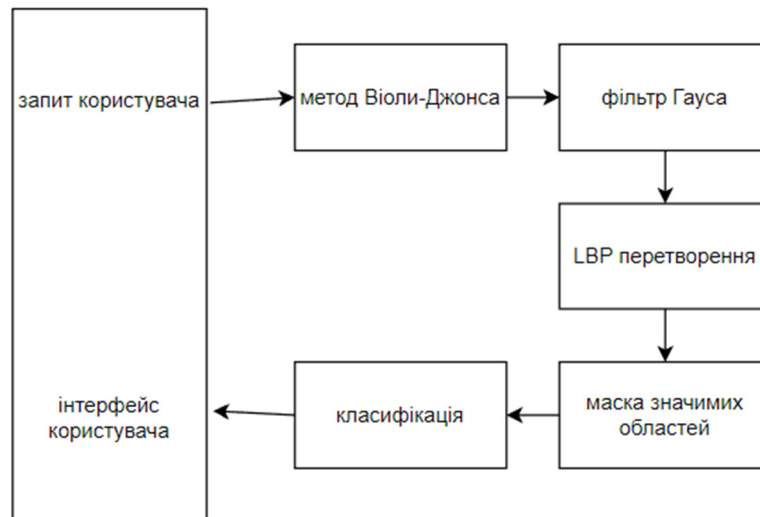


Рисунок 3.1 – Структурна схема модуля

Запит від користувача передається далі до блоку виявлення обличчя методом Віоли-Джонса де обличчя виявляються проте для оптимізації далі передається до блоку використання фільтра Гауса для оптимізації зображення далі передається в блок LBP перетворень де алгоритм проходиться по зображенню далі використовується маска значимих областей та класифікація. Отриманий результат в якості виявлених та виділених обличч виводиться на екран.

Для реалізації програмного модулю виявлення обличчя розроблено код на мові програмування python (Додаток А).

Також для розробки програмного модуля виявлення обличчя системою комп'ютерного зору розроблено діаграму станів що зображено на рисунку 3.2

						ДП.КН.9500032.058.ПЗ	Арк.
							41
Зм.	Арк.	№ докум.	Підпис	Дата			

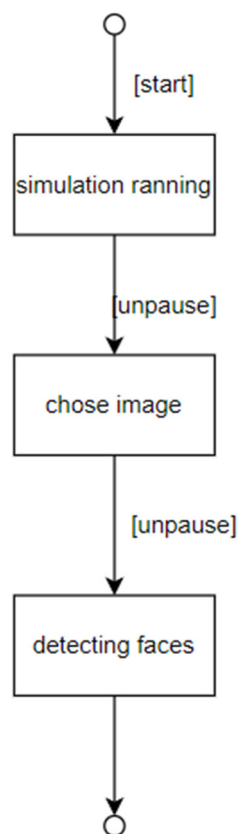


Рисунок 3.2 - Діаграма станів програмного додатку

Як видно з рисунку 3.2, після початку роботи програми програма очікує на дії з боку користувача після дії користувача, а саме вибору зображення. Далі програма знову зупиняється та очікує наступної дії від користувача, а саме дії виявити обличчя на зображенні після чого програма знову чекає на подальші дії користувача.

3.2 Інтерфейс користувача

Для розробки програмного модулю виявлення обличч було розроблено графічний інтерфейс користувача за допомогою бібліотеки python Tkinter [16].

Програма починає роботу з основного вікна де присутні дві кнопки «Завантажити зображення» та «Виявити обличчя» (Рисунок 3.3).

						ДП.КН.9500032.058.ПЗ	Арк.
							42
Зм.	Арк.	№ докум.	Підпис	Дата			

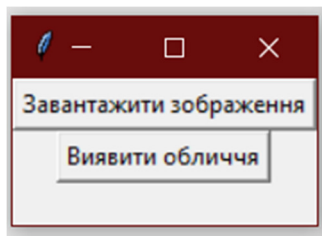


Рисунок 3.3 – Основне вікно модуля.

Далі після вибору зображення, для прикладу було взято зображення з інтернету, графічний інтерфейс додатку дещо модифікується – розширюється відносно розмірів зображення (представлено на рисунку 3.4).

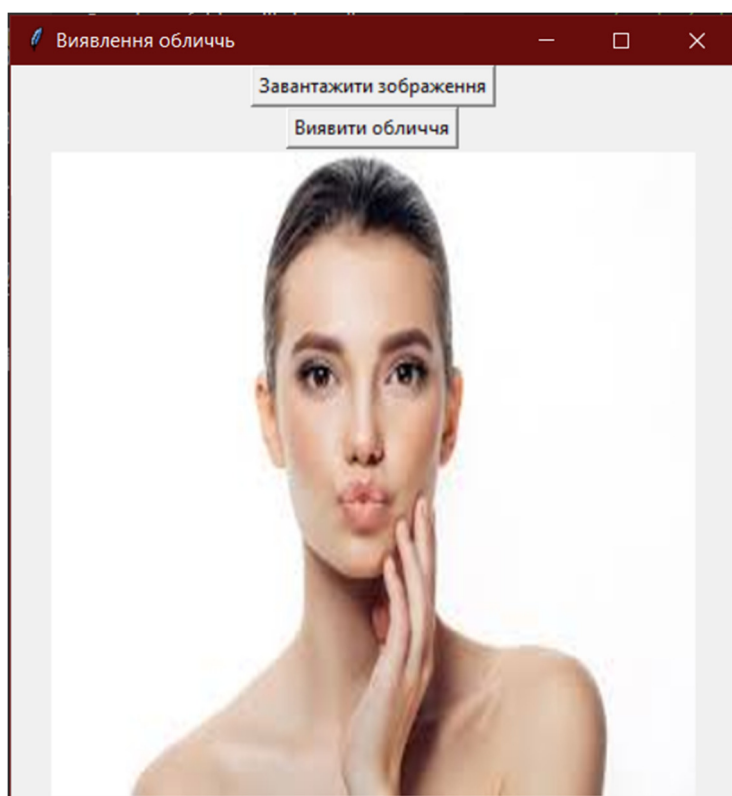


Рисунок 3.4 –Графічний інтерфейс основного вікна модуля.

Далі після натиску на кнопку «Виявити обличчя» буде показано прямокутник навколо обличчя, як зображено на рисунку 3.5. Зображення трішки розмилося в наслідок використання фільтра Гауса.

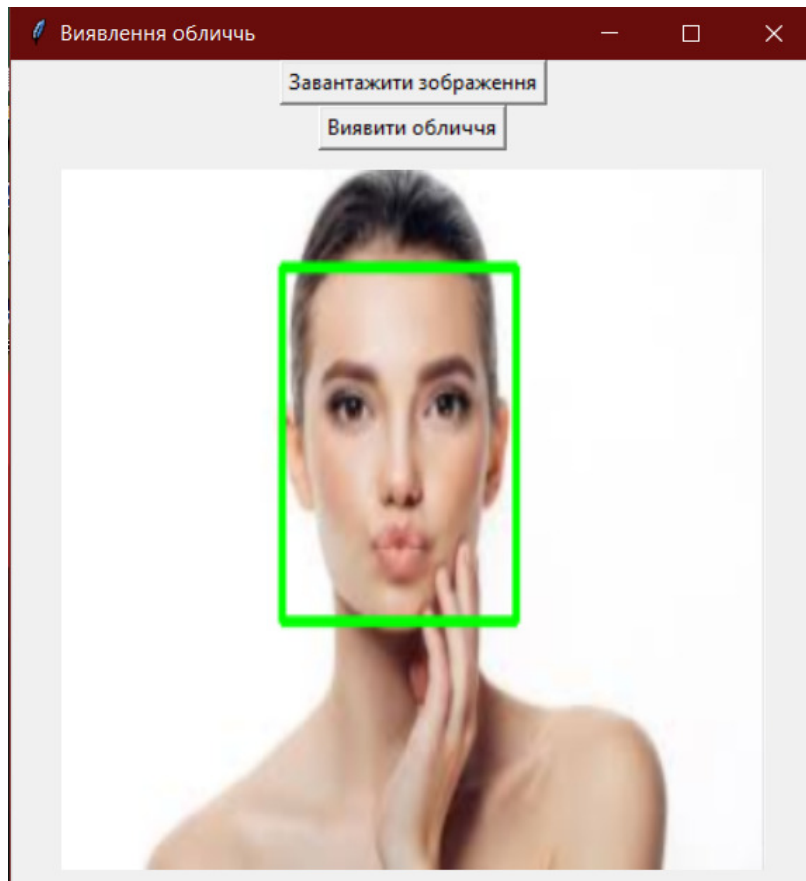


Рисунок 3.5 – Виявлене обличчя.

3.3 Тестування програмного забезпечення

Тестування програмного забезпечення дозволяє перевірити та оцінити програмне забезпечення з метою забезпечення його якості та відповідності вимогам [17].

Цей процес включає в себе запуск програмного забезпечення з різними вхідними даними та використанням різних функцій для перевірки та оцінки правильності роботи програми. Тестування ПЗ може бути проведене як вручну, так і за допомогою автоматизованих засобів. В ручному тестуванні, тестувальник запускає програму, виконує різні дії та перевіряє результати. У автоматизованому тестуванні, створюються скрипти тестування, які автоматично запускають програму, виконують різні дії та перевіряють результати. Тестування ПЗ допомагає виявляти помилки та недоліки програми

					ДП.КН.9500032.058.ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

перед її випуском. Це дозволяє вирішувати проблеми та вдосконалювати програмне забезпечення для забезпечення його якості та надійності. Тестування ПЗ допомагає також забезпечити відповідність програмного забезпечення вимогам та специфікаціям. Для успішного тестування ПЗ необхідно створити ефективні тестові сценарії, які охоплюють всі можливі випадки використання програмного забезпечення. Також важливо мати доступ до різноманітних тестових середовищ та забезпечувати ретельний аналіз результатів тестування. У цілому, тестування програмного забезпечення є важливою складовою процесу розробки програмного забезпечення та забезпечує якість та надійність програми.

Рівні тестування - це різні етапи тестування програмного забезпечення, які мають визначені цілі та завдання. В залежності від мети тестування, можуть використовуватися різні рівні тестування, кожен з яких має свої характеристики та особливості.

Основні рівні тестування програмного забезпечення:

- модульне тестування: це тестування окремих модулів програми (функціональних одиниць), зазвичай на ранніх етапах розробки. Модульне тестування дозволяє перевірити роботу кожного модуля окремо та виявити помилки та дефекти в ранній стадії.

- системне тестування: це тестування всієї системи програмного забезпечення в цілому, а не окремих модулів. Воно має на меті перевірити відповідність вимогам, стандартам та очікуванням клієнта.

- приймальне тестування: це тестування системи перед її введенням в експлуатацію або перед передачею замовнику. Воно має на меті перевірити, чи відповідає система специфікації, функціональним та нефункціональним вимогам, та чи задовольняє очікуванням користувачів.

- регресійне тестування: це тестування, яке проводиться після внесення змін у систему для перевірки того, що нові зміни не спричинили появу нових помилок.

					ДП.КН.9500032.058.ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Для тестування програмного модулю системи виявлення облич було обрано декілька видів та методів тестування програмного забезпечення на перевірку якості програмного продукту, а саме:

- модульне тестування;
- системне тестування;
- приймальне тестування;
- регресійне тестування.

Під час проведення модульного тестування програмного продукту було протестовано кожен окремий модуль на факт що він працює коректно та без критичних помилок та виконує задані йому функції. Було перевірено модуль виявлення облич методом Віоли-Джонса на наявність критичних помилок [18].

Було протестовано модуль на наборі даних бази зображень облич лабораторії Кембриджського Університету. Зображення одного обличчя під різними ракурсами представлені на рисунку 3.6.

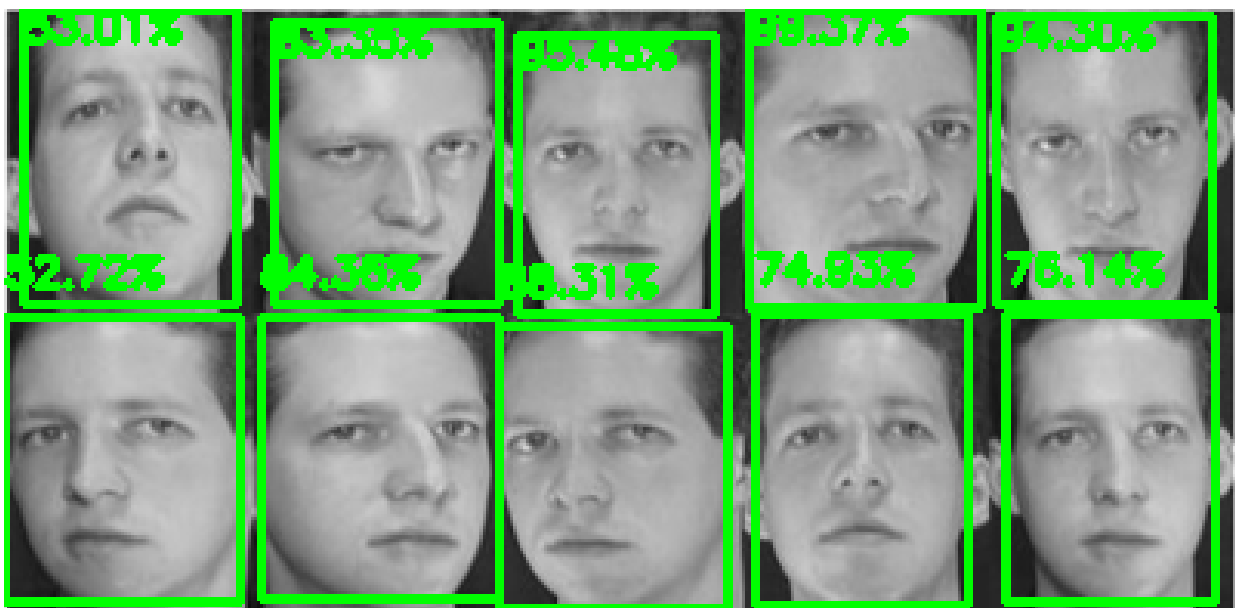


Рисунок 3.6 - Зображення з бази даних Кембриджського університету.

Результатом модульного тестування на цьому наборі зображень становить від 52.72 % до 89.37% результативності виявлення облич системним модулем Віоли-Джонса.

Другий набір даних – база зображень обличч лабораторії Єльського університету. Приклад зображень з другої бази обличч наведений на рисунку 3.7.

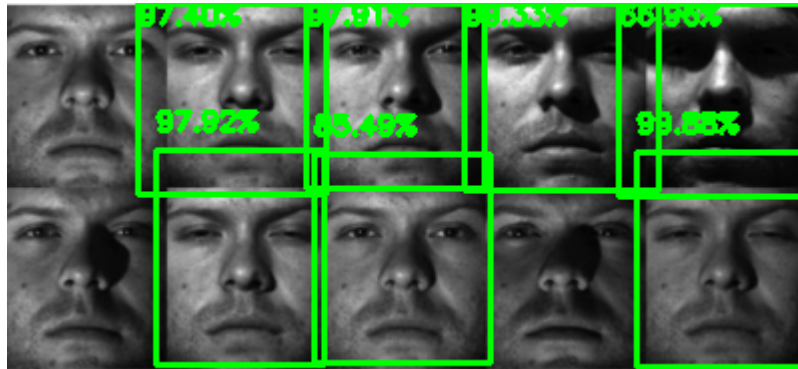


Рисунок 3.7 - Зображень обличч лабораторії Єльського університету

Результатом проведення модульного тестування системного модулю системи виявлення обличч було виявлення обличч високою результативністю 99.88%, проте деякі з обличч не були виявлені. Зменшення результативності виявлення обличч відбулося на підставі неоднорідності освітлення в даному наборі зображень.

Наступним кроком в комплексному тестуванні програмного продукту програмного модулю системи виявлення обличч було тестування фільтра Гауса на наявність критичних помилок багів та функціональності модулю системи виявлення обличч. Результатом модульного тестування фільтра Гауса на зображеннях з бази даних з бази даних Кембриджського університету становить 96.8% зображень вдалося оптимізувати та очистити від шумів, основною проблемою в оптимізації решти зображень становило майже не використовувані та застарілі формати зображення на усіх сучасних форматах зображення алгоритм виконував свої функції також критичних помилок не виявлено, баги було знайдено та усунено, основний баг полягав в тому що алгоритм фільтра Гауса повністю розмивав зображення, що робило неможливим виявлення обличчя в програмному модуль виявлення обличчя.

Також для тестування програного модулю виявлення обличч системою комп'ютерного зору було проведено додаткове тестування алгоритму. Для

					ДП.КН.9500032.058.ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

тестування було взято зображення з двома обличчями на фото для тестування алгоритму на предмет виявлення декількох обличчя на фото. Фото з двома обличчями представлено на рисунку 3.8.

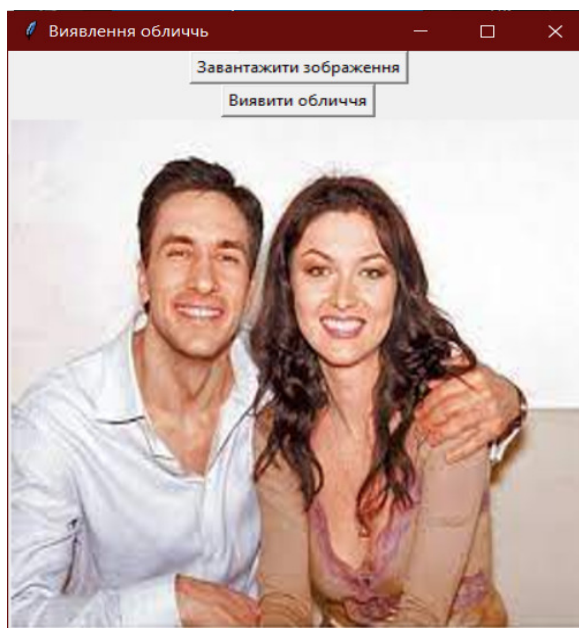


Рисунок 3.8 – Фото з двома людьми

Після обробки алгоритмом виявлення обличчя алгоритм повернув наступне зображення зображеного на рисунку 3.9.

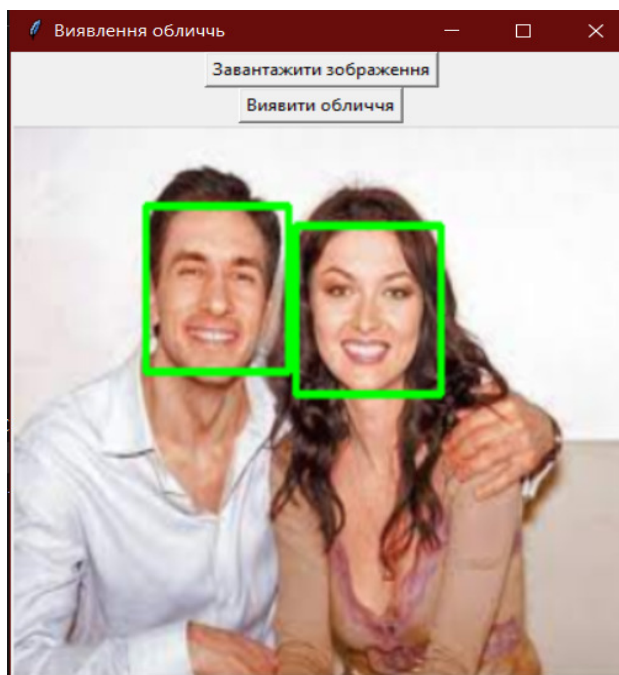


Рисунок 3.9 – Результат роботи програмного модуля

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Протестуємо розроблений програмний модуль на зображенні з декількома обличчями, зображення з декількома обличчями зображено на рисунку 3.10.



Рисунок 3.10 – Фото з декількома обличчями

Нажаль, алгоритм не виявив обличчя на фото тому, що обличчя були повернуті та алгоритм не спромігся виявити основні риси обличчя такі як:

- Очі;
- Ніс;
- Рот.

Наступним зображенням було моє зображення з камери ноутбука (рисунок 3.11)

					ДП.КН.9500032.058.ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

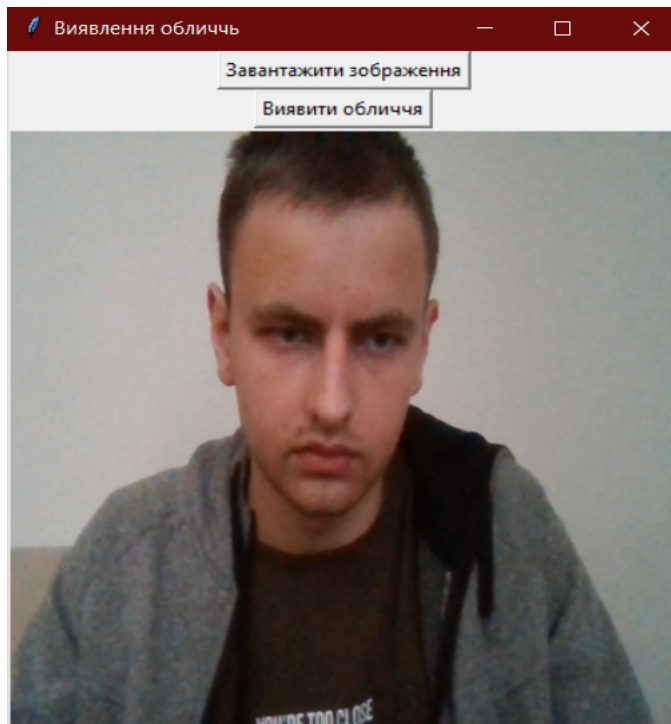


Рисунок 3.12 – фото без повороту голови

Як можна побачити з рисунку 3.12 фото взято з хорошим освітленням лице без повороту, кількість не обличч не занадто велике, то було дано фото рисунку 3.12 алгоритмові, результат представлено на рисунку 3.13.

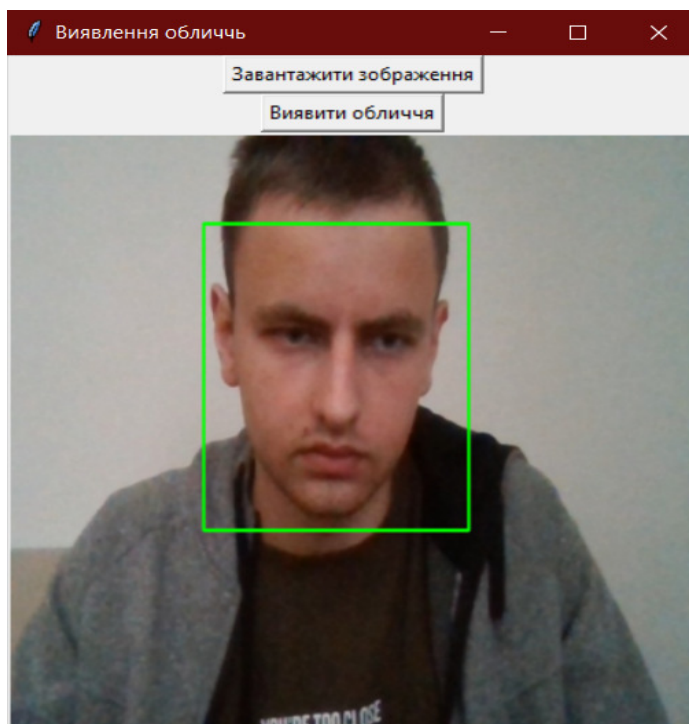


Рисунок 3.14 – Результат обробки алгоритмом

					ДП.КН.9500032.058.ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, під час тестування програмного забезпечення програмного модулю виявлення обличчя системою комп'ютерного зору було досягнуто наступних цілей: було протестовано систему на предмет критичних помилок та було протестована систему в різних ситуаціях та виявлено основні мінуси алгоритму. Загалом стан програмного забезпечення задовільний.

					ДП.КН.9500032.058.ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході дипломної роботи була проведена розробка модуля виявлення облич для системи комп'ютерного зору. Розроблена система має на меті автоматичне виявлення облич у зображеннях та відео з високою точністю та швидкістю.

У процесі розробки системи були вивчені та застосовані різні методи та алгоритми виявлення облич. Зокрема, були використані алгоритми обробки зображень, класифікатори та нейронні мережі. Використання цих методів дозволило досягти високої точності та швидкості виявлення облич. Розроблена система була протестована на різних типах зображень та відео з різними умовами освітлення.

Експериментальні результати підтвердили ефективність розробленої системи, її здатність точно виявляти обличчя навіть у складних умовах.

Розроблена система має широкі можливості застосування у різних сферах, таких як безпека, відеоспостереження, автоматичне розпізнавання осіб, реклама та розваги. Вона може бути використана для автоматизації процесів, пов'язаних з ідентифікацією та виявленням осіб у реальному часі. Проте, слід враховувати, що розроблена система має свої обмеження. Умови освітлення та якість вхідних даних можуть впливати на її точність та стабільність. Додаткові дослідження та оптимізація можуть покращити ефективність системи.

					ДП.КН.9500032.058.ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комп'ютерний зір [Electronic resource] – Mode of access: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B8%D0%B9_%D0%B7%D1%96%D1%80 - Title from the screen.
2. Amazon Rekognition pricing [Electronic resource] – Mode of access: <https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html> – Title from the screen.
3. FaceNet pricing [Electronic resource] – Mode of access: <https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/> – Title from the screen.
4. Kairos pricing [Electronic resource] – Mode of access: <https://worldvision.com.ua/kairos-zapuskaet-biometricheskoe-rasshirenie-tehnologii-raspoznavaniya-lits-dlya-roznichnykh-magazinov/> – Title from the screen.
5. Face API pricing [Electronic resource] – Mode of access: <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/> – Title from the screen.
6. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. – Харків: ХНУРЕ, 2021. – 124 с.
7. Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. - Paperback – January 23, 2018. – 305 с.
8. Остюченко А.О. Основи програмування мовою Python: навчальний посібник. - Ч.: ФОП Баликіна С.М., 2020. - 180 с.
9. V Kishore Ayyadevara, Yeshwanth Reddy. Modern Computer Vision with PyTorch: Explore deep learning concepts and implement over 50 real-world image applications 2020, 824 с.
10. Mohamed Elgendy Deep Learning for Vision Systems, 2020, 480 с.

					ДП.КН.9500032.058.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

11. Joseph Howse , Joe Minichino. Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning, 3rd Edition 2020, 372 c.

12. Valliappa Lakshmanan, Martin Görner, Ryan Gillard. Practical Machine Learning for Computer Vision. 2021. 274 c.

13. Anirudh Koul, Siddha Ganju, Meher Kasam. Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow 2019. 618 c.

14. Sunila Gollapudi. Learn Computer Vision Using OpenCV. 2019. 456 c.

15. Anand Deshpande, Vania V. Estrela, Navid Razmjoo. Computational Intelligence Methods for Super-Resolution in Image Processing Applications 2021. 573 c.

16. E. R. Davies, Matthew Turk. Advanced Methods and Deep Learning in Computer Vision. 2021. 582 c.

17. Steve Holden. Computer Vision: Advanced Techniques and Applications 2019. 215 c.

18. Mahmoud Hassaballah, Ali Ismail Awad. Deep Learning in Computer Vision Principles and Applications 2020. 643 c.

19. Xin Pan , Xuanhe Zhao , Weihong Yan , Jiangping Liu , Xiaoling Luo , Tana Wuyun. Computer Vision based Identification and Mosaic of Gramineous Grass Seeds 2021. 563 c.

20. Van Vung PhamH. ands-On Computer Vision with Detectron2: Develop object detection and segmentation models with a code and visualization approach 2023. 318 c.

21. Методичні вказівки до виконання дипломного проекту освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» / Укл. А.О. Саченко, М.П. Комар, Н.М. Васильків, Г.М. Гладій, В.С. Коваль. – Тернопіль: ЗУНУ, 2021. – 56 с.

					ДП.КН.9500032.058.ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А

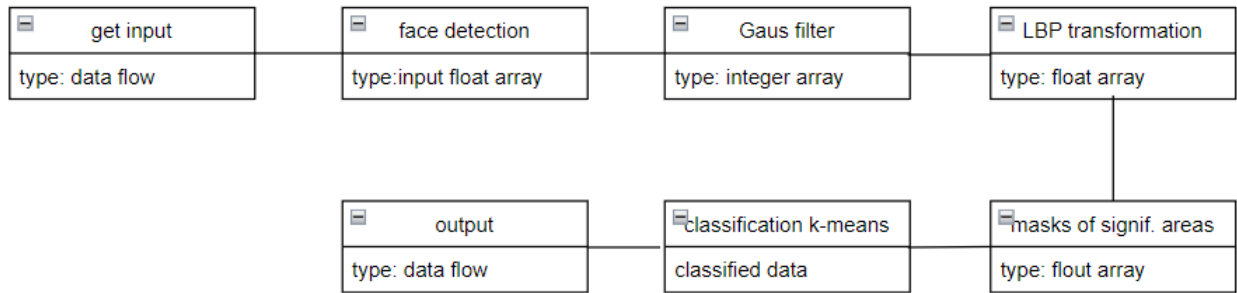
Алгоритм роботи модуля виявлення облич



Зм.	Арк.	№ докум.	Підпис	Дата

Додаток Б

Діаграма класів системи виявлення облич



Зм.	Арк.	№ докум.	Підпис	Дата

ДП.КН.9500032.058.ПЗ

Арк.

56

Додаток В Код модуля

```
import tkinter as tk
from tkinter import filedialog
from PIL import ImageTk, Image
import cv2
import numpy as np

# Функція, що відкриває вікно вибору файлу та завантажує зображення
def load_image():
    filename = filedialog.askopenfilename()
    global img
    img = Image.open(filename)
    img = img.resize((400, 400))
    global imgTk
    imgTk = ImageTk.PhotoImage(img)
    panel.configure(image=imgTk)
    panel.image = imgTk

# Функція, що виконує алгоритм виявлення обличчя на зображенні
def detect_faces():
    image = cv2.imread(str(img))
    # код для виконання алгоритму Віюлі-Джонса на завантаженому зображенні
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Завантаження каскаду класифікаторів
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    # Виявлення обличчя за допомогою каскаду класифікаторів
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
    # Застосування фільтра Гауса з ядром розміром 5x5 і сигмою 0.5
    gaussian_image = cv2.GaussianBlur(image, (5, 5), 0.5)
    # Define LBP parameters
    radius = 1
    n_points = 8 * radius

    # Apply LBP transform
    lbp = np.zeros_like(img)
    for i in range(radius, img.shape[0] - radius):
        for j in range(radius, img.shape[1] - radius):
            center = img[i, j]
            code = 0
```

					ДП.КН.9500032.058.ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

```

for n in range(n_points):
    x = i + int(round(radius * np.cos(2 * np.pi * n / n_points)))
    y = j - int(round(radius * np.sin(2 * np.pi * n / n_points)))
    if img[x, y] >= center:
        code += 2 ** (n_points - n - 1)
    lbp[i, j] = code
# Завантажуємо зображення та маску
mask = cv2.imread('mask.jfif', 0)
# Використовуємо маску для виділення значущих областей зображення
masked_img = cv2.bitwise_and(gaussian_image, gaussian_image, mask=mask)
# Виділяємо області з обличчями та зберігаємо їх у масиві
face_regions = []
for (x,y,w,h) in faces:
    face_regions.append(gray[y:y+h, x:x+w])
# Застосовуємо K-means для кластеризації значень пікселів областей обличчя
k = 2 # кількість кластерів
for region in face_regions:
    # Підготовка даних для кластеризації
    pixel_values = region.reshape((-1, 1))
    pixel_values = np.float32(pixel_values)
    # Виконуємо кластеризацію методом K-means
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    _, labels, centers = cv2.kmeans(pixel_values, k, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
    # Замінюємо значення пікселів у вихідному зображенні на значення кластерів
    centers = np.uint8(centers)
    res = centers[labels.flatten()]
    res2 = res.reshape((region.shape))
# Відображення знайдених облич у вікні зображення
for (x, y, w, h) in faces:
    cv2.rectangle(gaussian_image, (x, y), (x+w, y+h), (0, 255, 0), 2)
# Створення вікна та кнопок
root = tk.Tk()
root.title("Виявлення обличчя")
load_button = tk.Button(root, text="Завантажити зображення", command=load_image)
detect_button = tk.Button(root, text="Виявити обличчя", command=detect_faces)
panel = tk.Label(root)
# Розміщення кнопок та зображення на вікні
load_button.pack(side="top")
detect_button.pack(side="top")
panel.pack(side="bottom", fill="both", expand="yes")
# Запуск вікна
root.mainloop()

```

					ДП.КН.9500032.058.ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		