

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

ФЕДЧИШИН Вікторія Сергіївна

Програмний модуль виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору / Software Module for Feature Extraction from Images to Improve Classification Accuracy in Computer Vision Systems

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

Кваліфікаційна робота

Виконала студентка групи КН-42
В.С. Федчишин

Науковий керівник:
к.т.н., доцент П.Є. Биковий

Кваліфікаційну роботу допущено до захисту

«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н. М. Васильків

Тернопіль – 2025

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
В.о. завідувача кафедри
_____ Н.М. Васильків
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ФЕДЧИШИН Вікторія Сергіївна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмний модуль виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору / Software Module for Feature Extraction from Images to Improve Classification Accuracy in Computer Vision Systems

керівник роботи Биковий П.Є. к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- проаналізувати сучасні методи виділення ознак із зображень та оцінити їх ефективність у задачах класифікації в системах комп'ютерного зору;
- розробити концепцію побудови програмного модуля для багатокomпонентного виділення ознак із зображень;
- побудувати архітектуру модуля, що поєднує класичні методи (LBP, гістограми кольорів, вейвлет-перетворення, статистичні характеристики) з можливістю використання машинного навчання для класифікації;
- реалізувати програмний модуль із графічним інтерфейсом користувача, що дозволяє завантажувати зображення, витягувати ознаки та класифікувати їх;
- провести експериментальне дослідження ефективності модуля на відкритих датасетах із зображеннями облич, визначити точність, стабільність та адаптивність моделі;
- здійснити порівняльний аналіз точності класифікації при використанні різних комбінацій ознак і моделей машинного навчання.

5. Перелік графічного матеріалу в роботі:

- діаграма прецедентів
- діаграма послідовностей.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент _____ Федчишин В. С.
 (підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Биковий П.Є.
 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом у 76 сторінок і містить 8 ілюстрацій, 3 таблиці, 4 додатки та 31 використане джерело.

Метою кваліфікаційної роботи є створення програмного модуля для покращення точності класифікації зображень шляхом багатокomпонентного виділення ознак із подальшим використанням методів машинного навчання.

Об'єктом дослідження є процеси виділення ознак та класифікації зображень у системах комп'ютерного зору.

Розроблено програмний модуль виділення ознак зображень, що поєднує локальні бінарні шаблони (LBP), вейвлет-перетворення, гістограми кольорів і статистичні дескриптори. З метою зниження розмірності ознак реалізовано Principal Component Analysis (PCA). Для класифікації застосовано три алгоритми машинного навчання (SVM, Random Forest, Logistic Regression) та ансамблеву модель. Розроблений графічний веб-інтерфейс забезпечує завантаження зображень, обробку, класифікацію та візуалізацію результатів. Проведено експериментальну перевірку ефективності модуля та візуальний аналіз точності класифікації.

Ключові слова: ВИДІЛЕННЯ ОЗНАК, КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, PCA, LBP, ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ, АНСАМБЛЬ КЛАСИФІКАТОРІВ, ВЕБ-ІНТЕРФЕЙС.

ANNOTATION

Qualification work on the topic «Software module for feature extraction from images to improve classification accuracy in computer vision systems» for the degree of bachelor in specialty 122 «Computer Science» of the educational program «Computer Science» is written in 76 pages and contains 8 illustrations, 3 tables, 4 appendices and 31 references.

The purpose of the qualification work is to create a software module to improve the accuracy of image classification by multicomponent feature extraction with the subsequent use of machine learning methods.

The object of study is the processes of feature extraction and image classification in computer vision systems.

A software module for image feature extraction combining local binary patterns (LBP), wavelet transforms, color histograms, and statistical descriptors has been developed. Principal Component Analysis (PCA) is implemented to reduce the dimensionality of the features. For classification, three machine learning algorithms (SVM, Random Forest, Logistic Regression) and an ensemble model were used. The developed graphical web interface provides image uploading, processing, classification, and visualization of the results. Experimental verification of the module's efficiency and visual analysis of classification accuracy are carried out.

Keywords: FEATURE EXTRACTION, COMPUTER VISION, IMAGE CLASSIFICATION, PCA, LBP, WAVELET TRANSFORM, ENSEMBLE OF CLASSIFIERS, WEB INTERFACE.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області	10
1.1 Аналіз прикладної області	10
1.2 Опис об'єкту дослідження	12
1.3 Аналіз існуючих рішень та постановка задач проекту	13
2 Алгоритмічне та інформаційне забезпечення.....	20
2.1 Методи та алгоритми виділення ознак зображення	20
2.2 Розробка архітектури пропонованого рішення	23
2.3 Апаратна архітектура	26
2.4 Програмне забезпечення та алгоритми	27
2.5 Етапи розробки моделі машинного навчання.....	31
3 Програмна реалізація методу виділення ознак.....	37
3.1 Опис архітектури та можливостей методу	37
3.2 Навчання моделі методу виділення ознак	43
3.3 Оцінка ефективності методу виділення ознак	47
3.4 Реалізація графічного веб-інтерфейсу	53
Висновки.....	57
Список використаних джерел.....	59
Додаток А Програмна реалізація модуля витягування розширених ознак.....	63
Додаток Б Програмна реалізація модуля візуалізації результатів класифікації	68
Додаток В Програмна реалізація графічного веб-інтерфейсу для завантаження зображень та класифікації	70
Додаток Г Копія публікації.....	72

ВСТУП

Комп'ютерний зір є однією з найдинамічніших галузей штучного інтелекту, що активно розвивається і знаходить широке застосування у промисловості, медицині, безпеці, автономних системах, біометрії та мультимедіа. Одним із ключових етапів обробки візуальної інформації є виділення ознак – процес, який забезпечує перетворення «сирих» зображень у структуроване та інформативне представлення для подальшої класифікації чи розпізнавання.

Актуальність теми дослідження обумовлена стрімким зростанням обсягів візуальних даних у сучасному цифровому середовищі та необхідністю створення ефективних засобів їх автоматичної обробки. Традиційні методи виділення ознак, такі як SIFT, SURF або HOG, попри свою інформативність, часто поступаються в точності та продуктивності сучасним вимогам. Водночас методи на основі глибокого навчання, хоч і забезпечують високу якість результатів, мають високу обчислювальну складність. У зв'язку з цим важливо розробити програмний модуль, який поєднує переваги класичних і сучасних підходів до виділення ознак для підвищення точності класифікації при збереженні обчислювальної ефективності.

Задача якісного виділення ознак є особливо критичною для задач розпізнавання облич, де точність класифікації безпосередньо впливає на надійність систем контролю доступу, безпеки або біометричної ідентифікації. Ефективні системи повинні враховувати варіації освітлення, геометричні трансформації, мімічні зміни та інші фактори, що ускладнюють аналіз зображень у реальних умовах.

Метою цієї роботи є розробка програмного модуля виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору на основі комплексного підходу, що поєднує традиційні дескриптори та сучасні методи машинного навчання.

Завдання дослідження:

- провести аналіз сучасних методів виділення ознак зображень та існуючих програмних рішень;
- розробити алгоритмічну основу багатокomпонентного виділення ознак;
- побудувати архітектуру програмного модуля з використанням ансамблевого підходу до класифікації;
- реалізувати модуль із графічним інтерфейсом користувача;
- провести експериментальне дослідження ефективності модуля на відкритих наборах даних;
- здійснити порівняльний аналіз результатів з існуючими рішеннями.

Об’єкт дослідження – процеси виділення ознак і класифікації зображень у системах комп’ютерного зору.

Предмет дослідження – методи, алгоритми та програмні засоби багатокomпонентного виділення ознак для підвищення точності класифікації.

Методи дослідження включають аналіз наукових джерел, математичне моделювання, розробку та тестування алгоритмів обробки зображень, експериментальні дослідження на еталонних датасетах, статистичну оцінку результатів та порівняльне тестування.

Технічна і програмна реалізація модуля здійснювалася мовою програмування Python з використанням таких бібліотек: OpenCV (обробка зображень), PyWavelets (вейвлет-перетворення), scikit-learn (класифікація), scikit-image (виділення текстурних ознак), NumPy (числові обчислення) та Matplotlib (візуалізація).

Наукова новизна роботи полягає у створенні комплексного підходу до виділення ознак, який поєднує вейвлет-аналіз, локальні бінарні патерни, статистичні дескриптори та колірні гістограми в єдиний багатовимірний простір ознак, що дозволяє використовувати ефективні ансамблеві методи класифікації.

Практична цінність полягає у створенні готового програмного модуля, придатного до інтеграції в різноманітні системи комп’ютерного зору для підвищення точності розпізнавання. Згідно з результатами тестування, модуль забезпечує покращення точності класифікації на 15–20 % порівняно з підходами, що використовують лише окремі типи ознак.

Сфери застосування результатів: системи контролю доступу, безпеки, відеоспостереження, біометричної ідентифікації, медичної діагностики, промислової автоматизації, мультимедійної аналітики.

Результати дослідження опубліковано в матеріалах науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІТАР – 2025), м. Тернопіль, 27–29 травня 2025 р. (додаток Г).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз прикладної області

У сучасному цифровому світі системи комп'ютерного зору стали ключовим інструментом у розвитку інтелектуальних технологій. Вони використовуються в численних галузях: охорона правопорядку, медицина, виробництво, «розумні» міста, безконтактні платформи доступу тощо. Одним із найважливіших завдань у цій галузі є розпізнавання людей на зображеннях, що має широке застосування – від відеоспостереження та контролю доступу до ідентифікації осіб у криміналістиці чи персоналізації сервісів [1-4].

Методи розпізнавання людей стикаються з низкою викликів: варіативність освітлення (день, ніч, туман), зміна ракурсу, часткове перекриття обличчя, різноманітність зовнішності, віку, міміки, а також обмеження у кількості якісних даних для навчання моделей [1-3]. Додаткові труднощі виникають через візуальні спотворення, викликані геометричними трансформаціями, шумом або низькою якістю зображень, отриманих з камер спостереження.

Значним проривом у цій сфері стало впровадження глибокого навчання, зокрема згорткових нейронних мереж (CNN), які здатні автоматично виділяти ознаки зображення на різних рівнях абстракції [5-7]. Такі моделі, як FaceNet, DeepFace, MTCNN та ResNet, дозволили досягти високої точності ідентифікації навіть у складних умовах. Їх перевага полягає у здатності одночасно локалізувати обличчя, нормалізувати зображення, витягувати вектор ознак (embedding) та порівнювати його з еталонною базою.

Водночас, незважаючи на високі результати, існує потреба у вдосконаленні модулів виділення ознак, які могли б працювати ефективніше у випадках, коли дані мають обмежений обсяг або низьку якість. Традиційні методи, такі як локальні бінарні шаблони (LBP), гістограми орієнтованих градієнтів (HOG), вейвлет-перетворення, залишаються актуальними для доповнення або попередньої обробки зображень [8].

Окрему роль у сучасних підходах відіграють генеративні адверсаріальні мережі (GAN), які дозволяють створювати синтетичні зображення для

підвищення обсягу навчального набору або відновлення частково втрачених фрагментів зображення [9]. Крім того, активно розвиваються методи підвищення роздільної здатності (super-resolution), автоматичного доповнення даних, а також гібридні системи, що поєднують візуальні модулі з технологіями доповненої (AR) і віртуальної реальності (VR) [8].

За даними Gonzalez і Woods [10], попередня обробка та трансформація зображень є базовими етапами цифрової обробки, що дозволяють покращити якість вхідних даних для інтелектуальних алгоритмів. У свою чергу, згідно з дослідженнями O'Shaughnessy [11], виділення релевантних ознак є критичним етапом у побудові систем машинного навчання, оскільки саме від якості ознак залежить здатність моделі ефективно відокремлювати класи в багатовимірному просторі. Таким чином, у задачах класифікації зображень доцільним є створення спеціалізованих модулів, які забезпечують розширене виділення дескрипторів на основі поєднання статистичних, просторово-частотних та кольорових характеристик.

Актуальність цієї прикладної області визначається не лише технологічним прогресом, але й соціальним запитом на безпечні, точні та адаптивні системи ідентифікації. Наприклад, системи контролю на кордоні, міський відеомоніторинг, банківські сервіси або безконтактні методи авторизації вимагають високої точності при мінімальних помилках спрацювання.

Таким чином, розробка програмного модуля виділення ознак зображень для покращення точності класифікації є важливим напрямком досліджень, що поєднує передові досягнення у галузі комп'ютерного зору, обробки зображень і штучного інтелекту. Використання таких модулів у задачах відеоспостереження дозволяє забезпечити розпізнавання осіб у режимі реального часу з урахуванням змінних зовнішніх умов.

1.2 Опис об'єкту дослідження

У межах даного дослідження об'єктом є програмний модуль виділення ознак із зображень для підвищення точності класифікації в системах комп'ютерного зору. Такий модуль включає етапи попередньої обробки, побудови дескрипторів, формування векторів ознак та їх подальшої класифікації із застосуванням методів машинного навчання.

У сфері комп'ютерного зору, особливо в задачах розпізнавання об'єктів і класифікації зображень, процес виділення інформативних ознак є фундаментальним етапом. Саме якість і релевантність обраних ознак визначає здатність моделі ефективно розрізняти класи у багатовимірному просторі ознак, що прямо впливає на точність класифікації. Залежно від типу даних, умов зйомки, рівня шуму або варіативності зображень, ефективність тієї чи іншої техніки може суттєво відрізнятись. Тому дослідження та вдосконалення методів виділення ознак є ключовим напрямком у розробці сучасних систем візуального розпізнавання.

Як зазначає Худонг Ян [12], процес виділення ознак є найбільш складним і водночас критичним кроком у системах комп'ютерного зору. Він вимагає знаходження найбільш дискримінативних характеристик зображення, що дозволяють точно ідентифікувати об'єкти в умовах реального світу. Методи виділення ознак умовно поділяються на чотири категорії:

- методи на основі знань експертів;
- підходи, засновані на локальній структурі зображення;
- методи, що враховують глобальну структуру зображення;
- статистичні підходи, засновані на машинному навчанні [12].

Ці методи активно застосовуються у біометричних системах, зокрема у верифікації відбитків пальців, розпізнаванні облич, ідентифікації райдужки ока тощо. Кожен підхід має свої сильні сторони та обмеження, зумовлені складністю реальних сценаріїв – таких як варіації освітлення, пози, виразів обличчя або наявність шумів.

Особливу увагу сучасні дослідники приділяють масштабованості обчислень при роботі з великими наборами зображень. Так, у дослідженні [13] великомасштабної класифікації зображень на основі ImageNet запропонували систему Nadoor для паралельного виділення складних ознак із використанням сотень вузлів. Це дозволило обробити 1.2 мільйона зображень за один день. Для навчання класифікаторів вони застосували алгоритм ASGD (Averaged Stochastic Gradient Descent), який показав високу ефективність при роботі з терабайтами даних, забезпечивши швидке збіження моделі.

Додатково, як зазначає у роботі [14], побудова автоматизованої системи розпізнавання цілей для таких галузей, як відеоспостереження та безпілотні системи, базується на трьох основних етапах: виділення ознак (HOG, CNN), класифікація (SVM, Random Forest) і кластеризація (SSIM). Такий підхід дозволяє ефективно обробляти великі обсяги зображень та виконувати точне розпізнавання об'єктів.

Таким чином, інтеграція класичних методів виділення ознак (наприклад, LBP, HOG, вейвлет-перетворення) із сучасними технологіями (CNN, GAN, супер роздільність, ASGD) дозволяє будувати адаптивні та продуктивні модулі, які демонструють високу точність навіть в умовах реального часу та обмежених ресурсів.

1.3 Аналіз існуючих рішень та постановка задач проекту

Існуючі програмні рішення у сфері комп'ютерного зору демонструють значний прогрес у виявленні та класифікації об'єктів на зображеннях та відео. Різноманітні алгоритми машинного навчання, зокрема глибокі згорткові нейронні мережі (CNN), успішно застосовуються для вирішення завдань розпізнавання об'єктів, класифікації сцен, трекінгу та семантичної сегментації. Попри досягнення у точності класифікації, перед сучасними системами все ще постають виклики, пов'язані з обмеженнями в обчислювальних ресурсах, чутливістю до шуму, освітлення, позиції об'єктів і узагальненням моделей для нових сценаріїв.

Одним із найпоширеніших інструментів для обробки зображень у дослідницькому та промисловому середовищі є OpenCV (Open Source Computer Vision Library). Ця бібліотека надає багатий набір функцій для попередньої обробки зображень, детекції об'єктів, трекінгу, виділення ознак (таких як Наар-каскади, SIFT, SURF, ORB, HOG), морфологічних операцій та фільтрації [15]. OpenCV також забезпечує сумісність із іншими бібліотеками, такими як scikit-learn і TensorFlow, що дозволяє інтегрувати класичні та нейронні методи у єдину обчислювальну систему [16]. У багатьох практичних реалізаціях класичні ознаки з OpenCV використовуються для початкового аналізу сцени, попередньої класифікації або як вхідні дані для машинного навчання.

Історично традиційні методи виділення ознак включали детекцію контурів, гістограми напрямків градієнтів (HOG), а також локальні дескриптори, як-от SIFT і SURF. Ці методи є стійкими до змін масштабу та повороту, але можуть втрачати ефективність при сильних спотвореннях, зміні освітлення чи зашумленості. Поява глибоких згорткових нейронних мереж (CNN) змінила підхід до виділення ознак – тепер ознаки автоматично навчаються всередині нейронної архітектури, що дозволяє досягати вищої точності, особливо на великих обсягах даних.

Серед сучасних підходів до обробки зображень особливе місце займає архітектура YOLO – потужна CNN-модель, здатна розпізнавати об'єкти на відеопотоці з високою швидкістю [17]. YOLOv8 підтримує кілька варіантів розміру (Nano, Small, Medium, Large, Extra Large), що дозволяє адаптувати модель до апаратних можливостей. У цьому випадку використано варіант Medium (YOLOv8m), який забезпечує оптимальне співвідношення між швидкістю обробки (понад 30 кадрів за секунду на GPU) і точністю розпізнавання, враховуючи характеристики апаратного забезпечення [18]. Порівняння характеристик різних версій YOLO наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння версій YOLO

Версія	Точність (mAP@0.5)	Швидкість (FPS, GTX 1080)	Кількість параметрів	Примітка
YOLOv5s	~0.85	55-65	~7m	Швидка, але менш точна
YOLOv8n	~0.87	65-75	~3m	Легка, для слабкого GPU
YOLOv8m	~0.92	40-45	~25m	Оптимальна для модуля
YOLOv8x	~0.95	25-30	~68m	Висока точність, повільна

У дослідженні [15] було застосовано модифіковану модель Residual SqueezeNet, орієнтовану на розпізнавання марки та моделі транспортного засобу за фронтальними зображеннями. Архітектура зменшувала обчислювальну складність, зберігаючи високу точність (96,3%), проте мала обмеження щодо ракурсів і сценаріїв використання.

У своїй роботі Лоу (2004) запропонував метод виявлення та опису ключових точок на зображеннях, відомий як SIFT (Scale-Invariant Feature Transform), який дозволяє виявляти стабільні ознаки, незалежні від масштабних змін, поворотів і частково – змін освітлення. Метод складається з кількох етапів: побудова масштабного простору для виявлення екстремумів, точне визначення положення ключових точок, відбір орієнтації та побудова дескрипторів. Завдяки інваріантності до геометричних і фотометричних перетворень, SIFT став фундаментальним інструментом для багатьох завдань комп'ютерного зору, зокрема розпізнавання об'єктів, зіставлення зображень та побудови мозаїк [19].

Основна ідея полягає в поєднанні кількох незалежних груп ознак, що формуються на основі обробки вхідного зображення. Зокрема, модуль включає такі ключові компоненти:

а) підсистема попередньої обробки зображення, яка виконує нормалізацію, зменшення шуму та зміну кольорового простору (з RGB в HSV), що забезпечує стійкість до освітлення;

б) блок виділення ознак, який формує дескриптори на основі:

- 1) вейвлет-перетворень (наприклад, Haar, Daubechies) [20];
- 2) локальних бінарних патернів (LBP);
- 3) гістограм кольорів;
- 4) статистичних характеристик (середнє, дисперсія, різкість);
- 5) сирих піксельних значень зменшених зображень;

в) компонент об'єднання ознак, який нормалізує та з'єднує всі дескриптори у єдиний вектор ознак;

г) модуль класифікації, що приймає вектор ознак як вхід і виконує класифікацію за допомогою нейронної мережі або іншого алгоритму машинного навчання (наприклад, SVM або KNN).

Для підвищення точності класифікації в системах комп'ютерного зору все більш актуальним стає поєднання різнорідних типів ознак – статистичних, частотних, текстурних та геометричних. Таке комбінування дозволяє отримати більш повне та стійке представлення зображення, що особливо важливо в умовах недостатньої кількості навчальних зразків. Зокрема, використання хвильового перетворення, локальних бінарних шаблонів (LBP), гістограм градієнтів (HOG) або інших класичних методів у поєднанні з сучасними підходами дозволяє значно покращити якість класифікації [21-23].

Дослідження підтверджують ефективність комбінованих підходів. Наприклад, поєднання статистичних та модельних текстурних ознак, таких як GMRF та fBm, забезпечує точність класифікації до 97,01 % [21]. Інше дослідження показало, що об'єднання текстурних та геометричних ознак покращує точність класифікації до 85,2 % при використанні k-NN [23].

Однією з ключових причин зниження ефективності систем розпізнавання є недостатньо інформативне або нестабільне виділення ознак – процес, у якому зображення перетворюється у формалізовані дані, що надалі подаються на вхід класифікатору. У випадках, коли використовуються лише «сірі» зображення або автоматичне витягування ознак у нейромережах без належної оптимізації, результат може бути незадовільним, особливо у випадках обмеженого набору даних або складних умов зйомки [24-26]. Це пов'язано з тим, що глибокі нейронні мережі, такі як DCNN, потребують великої кількості навчальних зразків для

ефективного навчання. При обмеженій кількості даних існує ризик перенавчання, що призводить до зниження продуктивності на нових даних.

Таким чином, поєднання різних типів ознак та використання комбінованих підходів до виділення ознак є перспективним напрямом для покращення точності класифікації в системах комп'ютерного зору, особливо в умовах обмежених обчислювальних ресурсів та недостатньої кількості навчальних даних. Класичні методи (OpenCV, SIFT, HOG, GMM) є ефективними при обмежених обчислювальних ресурсах або коли відсутня можливість використовувати глибоке навчання. Одним із ключових рішень стало використання вейвлет-перетворень, які дають змогу аналізувати зображення на різних масштабах та виявляти локальні особливості.

У поєднанні з дескрипторами, що ґрунтуються на LBP, та кольорними гістограмами, це забезпечує велику варіативність ознак для тренування класифікатора. У рамках аналізу програмних засобів виявлено такі основні підходи до виділення ознак:

- класичні дескриптори (HOG, LBP, SIFT, SURF, Haar), реалізовані в OpenCV і scikit-image. Ці методи дозволяють детально аналізувати текстуру, контур, форму об'єктів і є добре відпрацьованими у задачах, де важлива інтерпретованість та швидкодія;

- хвильовий аналіз (Wavelet Transform) – застосовується для багаторівневої декомпозиції зображень, дозволяє виділяти частотні компоненти, що можуть бути корисними при розпізнаванні локальних структур і текстур;

- глибокі ознаки з CNN, які автоматично навчаються через архітектури типу ResNet, VGG або YOLO. Незважаючи на високу точність, ці методи є менш прозорими в інтерпретації результатів, а також вимагають значних ресурсів на навчання.

Деякі існуючі системи обмежуються одним методом ознак, не використовуючи потенціал комбінованого підходу, який міг би покращити точність за рахунок багатоканального подання зображень. На основі аналізу сформульовано такі основні проблеми, які потребують вирішення:

- недостатня точність класифікації у складних умовах освітлення та шуму;

- обмеження традиційних дескрипторів у випадках високої варіативності зображень;

- висока вартість ресурсів при використанні лише нейронних мереж;
- недостатнє поєднання різних видів ознак (текстурних, частотних, статистичних).

З урахуванням вищезазначеного, мета цієї роботи полягає у розробці програмного модуля для виділення ознак зображень, який:

- комбінує декілька типів ознак: LBP, хвильові перетворення та статистичні характеристики (середнє, дисперсія, ентропія тощо);

- забезпечує високу інформативність і стабільність ознак для подальшої класифікації;

- сумісний із популярними класифікаторами (SVM, KNN, Random Forest тощо);

- адаптований до реальних умов використання, з можливістю роботи з обмеженими обчислювальними ресурсами;

- дозволяє об'єктивно оцінити вплив різних ознак на точність розпізнавання.

Таким чином, завдання проекту можна сформулювати наступним чином:

- а) провести експериментальне порівняння різних методів виділення ознак;
- б) реалізувати модуль, що об'єднує декілька типів ознак у єдине векторне представлення;

- в) підібрати ефективний класифікатор та навчити його на отриманих ознаках;

- г) провести тестування системи на відкритих наборах зображень;

- г) проаналізувати вплив кожного типу ознак на точність класифікації.

У результаті реалізований програмний модуль має стати універсальним інструментом для задач класифікації в системах комп'ютерного зору, з акцентом на простоту, ефективність та адаптивність до різних умов.

Таким чином, у вміє розділі було здійснено всебічний аналіз сучасних підходів до задач комп'ютерного зору, зосереджено увагу на перевагах і недоліках як традиційних методів обробки зображень, так і сучасних нейронних

моделей. Виявлено потребу у вдосконаленні етапу виділення ознак як ключового чинника підвищення точності класифікації. На основі проведеного аналізу сформульовано концепцію розробки програмного модуля, що поєднує різні типи ознак для забезпечення гнучкого, інтерпретованого та ефективного рішення в системах комп'ютерного зору.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Методи та алгоритми виділення ознак зображення

Для підвищення ефективності систем комп'ютерного зору в задачах розпізнавання облич важливо сформувати інформативні та стійкі ознаки, що точно характеризують вхідні зображення та забезпечують високу якість класифікації. Основною метою цього етапу є розробка методу виділення ознак, який дозволяє створювати репрезентативні вектори ознак для подальшого використання у класифікаційних моделях розпізнавання облич.

Для досягнення поставленої мети необхідно реалізувати такі завдання:

- розробити модуль, який комбінує декілька типів ознак (текстурних, статистичних, колірних) для побудови цілісного опису обличчя;
- забезпечити стійкість ознак до змін умов освітлення, масштабу та шумів, що є критичним для розпізнавання облич у різних умовах;
- підвищити точність класифікації шляхом включення локальних і глобальних характеристик обличчя;
- оптимізувати процес виділення ознак для використання у реальному часі або при обмежених обчислювальних ресурсах.

Математично задачу розпізнавання облич можна сформулювати наступним чином. Функція розпізнавання облич має вигляд:

$$f(I, O) = \{C, B\}, \quad (2.1)$$

де I – вхідне зображення;

O – параметри класифікаційної моделі;

C – ідентифікатор особи або клас обличчя;

B – координати обмежувальної рамки (bounding box) у форматі (x, y, w, h) (x, y – координати лівого верхнього кута, w, h – ширина та висота рамки навколо виявленого обличчя).

Функція втрат для оптимізації моделі визначається як:

$$L = L_{cls} + L_{detection}, \quad (2.2)$$

де L_{cls} – втрати класифікації, що вимірює точність ідентифікації особи;

$L_{detection}$ – втрати детекції, що оцінює точність виявлення обличчя та передбачення координат рамки.

Мета полягає в мінімізації L шляхом налаштування параметрів O під час навчання. Обмеження продуктивності має вигляд:

$$T_{proc} \leq \frac{1}{FPS_{min}}, \quad (2.3)$$

де T_{proc} – час обробки одного кадру;

FPS_{min} – мінімально необхідна частота кадрів за секунду.

Ці формулювання відображають основні аспекти задачі:

- точність розпізнавання облич;
- локалізація облич на зображенні;
- швидкість обробки у реальному часі.

У межах даного дослідження розроблено модель, яка реалізує функцію

$$f(I) \rightarrow O, \quad (2.4)$$

де I – вхідне зображення,

O – формалізоване представлення ознак обличчя,

що є вхідними даними для класифікатора. Навчання моделі спрямоване на мінімізацію функції втрат L , з урахуванням обмежень продуктивності, таких як обчислювальна складність та обсяг доступних ресурсів. Таким чином, задача полягає у створенні програмного модуля, який поєднує класичні підходи до виділення ознак зображень облич для покращення точності класифікації в умовах обмежених даних або ускладненого фону.

Запропонований метод базується на комбінуванні кількох різномірних груп ознак, що дозволяє отримати повніше представлення обличчя:

Вейвлет-ознаки (Haar, Daubechies DB4) використовуються для виявлення локальних текстур на різних масштабах обличчя. Завдяки своїй здатності аналізувати як просторову, так і частотну інформацію, вони є ефективними для багаторівневої фільтрації зображень облич. В реалізації використовуються вейвлети 'db4' та 'haar' з різними рівнями декомпозиції (2-3 рівні).

Локальні бінарні патерни (LBP) описують мікротекстуру обличчя, дозволяючи фіксувати дрібні варіації в структурі пікселів. Вони є стійкими до змін освітлення та широко застосовуються в задачах розпізнавання облич. В модулі використовується uniform LBP з радіусом 1 та 8 точками.

Гістограми кольорів у просторах RGB та HSV дозволяють враховувати колірну інформацію обличчя, яка може бути важливою при класифікації осіб із різними відтінками шкіри та характеристиками освітлення. Для кожного каналу обчислюється 16-бінна гістограма.

Статистичні параметри включають середнє значення, стандартне відхилення, мінімум, максимум яскравості, а також характеристики градієнтів (середнє та стандартне відхилення магнітуди градієнта) та варіацію лапласіана для оцінки різкості. Ці параметри відображають загальні властивості зображення обличчя та його контраст.

Зменшене зображення у вигляді сирих пікселів дає уявлення про глобальну форму та структуру обличчя. В реалізації використовується розмір 32×32 пікселі для оптимізації швидкості обробки.

Покращена попередня обробка включає нормалізацію освітлення за допомогою CLANE (Contrast Limited Adaptive Histogram Equalization), гауссове розмиття для зменшення шумів та масштабування до цільового розміру 64×64 пікселі.

Усі вказані ознаки підлягають нормалізації для збереження масштабної однорідності. Після цього вони об'єднуються в єдиний вектор ознак, який подається на вхід класифікаційній моделі, зокрема, алгоритмам машинного навчання (SVM, KNN, нейронні мережі).

Такий підхід дозволяє забезпечити оптимальний баланс між точністю, швидкістю та адаптивністю системи розпізнавання облич в умовах варіативного освітлення, фону та кількості даних.

2.2 Розробка архітектури запропонованого рішення

Розробка програмного модуля виділення ознак зображень облич спрямована на покращення точності класифікації та розпізнавання осіб у системах комп'ютерного зору. Традиційні підходи до розпізнавання облич часто страждають від недостатньої адаптивності до змін освітлення, ракурсу та мімічних виразів, а також від обмеженої здатності узагальнювати дані. Запропоноване рішення передбачає побудову багаторівневої архітектури, яка дозволяє ефективно виокремлювати релевантні ознаки зображень облич, зберігаючи при цьому високу продуктивність системи.

Основна ідея розробленого підходу полягає у поєднанні кількох незалежних груп ознак, що формуються в результаті багаторівневої обробки вхідного зображення обличчя. Така багатокомпонентна структура дозволяє врахувати як локальні характеристики (текстура шкіри, деталі очей, форма губ), так і глобальні особливості обличчя (загальна геометрія, пропорції), підвищуючи стійкість системи до варіацій освітлення, ракурсу, мімічних змін та масштабу.

Структура модуля виділення ознак облич включає чотири основні компоненти, кожен з яких виконує специфічні функції в процесі обробки зображень.

Перший компонент архітектури відповідає за попередню обробку та виявлення області обличчя. На цьому етапі зображення піддається низці операцій, спрямованих на покращення якості вхідних даних і локалізацію області обличчя. Процес включає детектування обличчя та очей з використанням каскадних класифікаторів Хаара для виявлення області обличчя та перевірки наявності двох очей, що гарантує коректність виявленої області. Далі здійснюється обрізання та масштабування з виділенням області обличчя з фіксованим розміром (наприклад, 64×64 пікселі) для забезпечення

уніфікованості вхідних даних. Важливим етапом є нормалізація освітлення через застосування адаптивної гістограмної еквалізації (CLAHE) для компенсації нерівномірного освітлення обличчя. Також проводиться зменшення шуму за допомогою застосування Гауссового фільтру для згладжування випадкових варіацій яскравості, зумовлених сенсором або освітленням, та перетворення кольорового простору з переходом із простору BGR до HSV, де компонента відтінку (Hue) є менш чутливою до зміни освітлення, що забезпечує більшу інваріантність до зовнішніх умов.

Другий компонент архітектури реалізує багатокomпонентне витягування ознак. Цей блок формує набір дескрипторів, які відображають різні аспекти структури та характеристик обличчя. Вейвлет-перетворення (Daubechies db4, Haar) дозволяє розкласти зображення обличчя на компоненти, які представляють як низькочастотну (загальну структуру обличчя), так і високочастотну (деталі та текстуру шкіри) інформацію. Використовуються різні типи вейвлетів на кількох рівнях розкладу для отримання багатшого представлення. Локальні бінарні патерни (LBP) кодують мікроструктуру навколо кожного пікселя, що особливо ефективно для опису текстури шкіри та локальних особливостей обличчя. Метод формує гістограму шаблонів, яка описує текстурні властивості на локальному рівні, забезпечуючи стійкість до варіацій яскравості та шуму. Гістограми кольорів описують розподіл кольорових пікселів у просторах BGR та HSV. Для кожного кольорового каналу будується гістограма з 16 бінами, що дозволяє відобразити домінуючі кольори обличчя та відтінки шкіри. Статистичні характеристики та градієнтні ознаки включають числові метрики, які узагальнюють структуру обличчя: середнє значення, стандартне відхилення, мінімум, максимум яскравості, а також характеристики градієнтів (оператори Sobel) та різкості (оператор Лапласа) для виявлення контурів та переходів на обличчі. Сирі піксельні значення зменшеного зображення формуються через масштабування зображення обличчя до фіксованого розміру (32×32), після чого всі його пікселі розгортаються в одномірний вектор. Цей вектор зберігає загальну форму та силует обличчя, будучи чутливим до його контурів та основних рис.

Третій компонент забезпечує нормалізацію та об'єднання ознак. Після формування окремих груп дескрипторів усі ознаки нормалізуються до уніфікованого масштабу шляхом ділення на 255 для піксельних значень та нормалізації гістограм. Далі вони конкатенуються в єдиний вектор ознак, який представляє повний опис обличчя у багатовимірному просторі. Така стратегія забезпечує комплексне представлення, придатне для подальшої класифікації та розпізнавання осіб.

Четвертий компонент архітектури являє собою класифікаційний блок. Формований вектор ознак подається на вхід класифікаційної моделі. У межах роботи передбачено використання нейронної мережі (MLP), яка здатна моделювати нелінійні залежності між ознаками обличчя та ідентичністю особи, методу опорних векторів (SVM), що є ефективним у високовимірному просторі ознак та забезпечує високу точність розпізнавання навіть за умов малої кількості навчальних прикладів для кожної особи, та методу к-ближчих сусідів (KNN), який є простим та інтерпретованим методом, що ґрунтується на метричних відстанях у просторі ознак обличчя.

Архітектура модуля є модульною та масштабованою, що дозволяє легко розширювати систему, додаючи нові методи виділення ознак або покращуючи наявні компоненти. Такий підхід створює гнучку платформу для побудови інтелектуальних систем розпізнавання облич, які можуть бути адаптовані під конкретні прикладні задачі – від систем контролю доступу до медичної діагностики.

Завдяки поєднанню класичних і сучасних методів виділення ознак, система демонструє високу точність класифікації облич навіть за умов обмеженої кількості навчальних даних, варіативного освітлення та різних ракурсів зйомки. Використання перевірки наявності двох очей як критерію якості виявленого обличчя підвищує надійність системи та зменшує кількість хибних спрацювань.

2.3 Апаратна архітектура

Апаратна архітектура програмного модуля, розробленого для виділення ознак зображень з метою покращення точності класифікації в системах комп'ютерного зору, враховує вимоги до високої обчислювальної продуктивності, масштабованості та ефективної обробки великого обсягу візуальних даних. Оскільки виділення ознак є критичним етапом у системах комп'ютерного зору, апаратна платформа має забезпечувати стабільну й швидку роботу модулів обробки та аналізу зображень.

Для реалізації модуля використано обчислювальну систему з такими характеристиками (таблиця 2.1):

- процесор: Intel Core i3-8100 із чотирма ядрами та тактовою частотою 3.60 GHz, що забезпечує базову обробку та управління задачами;
- оперативна пам'ять: 16 ГБ DDR4, яка дозволяє ефективно працювати з об'ємними зображеннями та наборами ознак;
- графічна карта: NVIDIA GTX 1080 з 8 ГБ відеопам'яті, що підтримує масово-паралельну обробку через CUDA та значно прискорює обчислення;
- накопичувач: твердотільний диск (SSD) із високою швидкістю читання/запису, що забезпечує швидкий доступ до зображень і проміжних результатів.

Таблиця 2.1 – Характеристики комп'ютера, середовища та інтерфейсу

Категорія	Інформація
Процесор (CPU)	Intel Core i3-8100 3.60GHz, 4 ядра
Відеокарта (GPU)	Nvidia GTX 1080 8gb
Оперативна пам'ять (RAM)	16gb ddr4
Операційна система	Windows 10
Бібліотеки	openCV,pytorch, tensorflow
Мова програмування	Python 3.11
Графічний інтерфейс	Gradio

Для досягнення високої якості класифікації, модуль використовує комбінацію ознак, які репрезентують важливі властивості зображень. До них належать:

- гістограми кольору в HSV-просторі – для опису кольорових характеристик;
- текстурні дескриптори (LBP) – для кодування локальних варіацій текстури;
- вейвлет-перетворення (Haar, Daubechies) – для багаторівневого опису зображення;
- статистичні характеристики – середнє, дисперсія, ентропія, тощо.

Основне обчислювальне навантаження лягає на графічний процесор, що дозволяє виконувати паралельну обробку зображень, особливо під час вейвлет-аналізу або згорткових операцій. Завдяки підтримці CUDA, система досягає високої продуктивності навіть при роботі з великими наборами даних.

Таким чином, запропонована апаратно-програмна архітектура є оптимальною для задач виділення ознак і класифікації зображень, забезпечуючи високу швидкодію, точність та гнучкість у налаштуванні й розширенні функціоналу системи комп'ютерного зору.

2.4 Програмне забезпечення та алгоритми

Програмний модуль, розроблений у межах даної роботи, спрямований на підвищення точності класифікації та розпізнавання облич у системах комп'ютерного зору шляхом глибокого аналізу зображень облич та багаторівневого виділення ознак. Основою реалізації модуля є мова програмування Python, яка забезпечує швидку розробку, підтримку модульності та широке використання потужних бібліотек для обробки зображень, аналізу даних і реалізації алгоритмів машинного навчання.

У роботі модуля використано сучасні інструменти обробки зображень, зокрема бібліотеки OpenCV, NumPy, PyWavelets, scikit-image, а також алгоритми машинного навчання для класифікації та розпізнавання облич. Таке середовище

дозволяє будувати гнучкі та масштабовані рішення, орієнтовані на застосування в реальних умовах із високими вимогами до точності та продуктивності розпізнавання осіб.

Модуль реалізує комбінований підхід до виділення ознак облич, що охоплює декілька незалежних груп дескрипторів, отриманих із різних рівнів обробки зображення обличчя. Такий підхід дозволяє враховувати як локальні характеристики обличчя (текстуру шкіри, деталі очей), так і глобальні особливості (загальну геометрію та пропорції).

Архітектура модуля включає такі компоненти:

а) підсистема детектування та попередньої обробки облич, яка здійснює:

1) детектування області обличчя за допомогою каскадних класифікаторів Хаара;

2) верифікацію якості виявленого обличчя через пошук двох очей;

3) обрізання та нормалізацію розмірів зображення обличчя до стандартного розміру (64×64);

4) покращення освітлення за допомогою адаптивної гістограмної еквалізації (CLAHE);

5) зменшення шумів через Гауссовий фільтр для згладжування артефактів зйомки;

б) перетворення кольорового простору (з BGR у HSV) для зменшення впливу освітлення на характеристики обличчя;

б) блок виділення ознак обличчя, що формує набір дескрипторів на основі:

1) вейвлет-перетворень (Daubechies db4, Haar) для багаторівневого аналізу структури обличчя з виділенням як загальної форми, так і дрібних деталей;

2) локальних бінарних патернів (LBP) для виділення текстурних характеристик шкіри та локальних особливостей обличчя;

3) статистичних ознак (середнє значення, стандартне відхилення, мінімум, максимум яскравості, характеристики градієнтів та різкості);

4) гістограм кольорів у просторах BGR та HSV для опису відтінків шкіри та загального колориту обличчя;

5) сирих піксельних значень зі зменшених зображень (32×32) для збереження загальної форми та силуету обличчя;

в) компонент об'єднання ознак, який забезпечує:

1) нормалізацію всіх векторів дескрипторів (ділення на 255 для піксельних значень, нормалізація гістограм);

2) конкатенацію усіх груп ознак у єдиний багатовимірний вектор фіксованої довжини;

3) створення комплексного представлення обличчя, придатного для класифікації;

г) модуль класифікації облич, який підтримує різні підходи:

1) класичні алгоритми машинного навчання, такі як SVM (Support Vector Machine), KNN (к-ближчих сусідів) для розпізнавання та верифікації осіб;

2) багатошарові перцептрони (MLP) для моделювання складних нелінійних залежностей між ознаками обличчя та ідентичністю особи.

Підсистема класифікації модульно налаштовується, що дає змогу досліджувати ефективність різних методів на одному наборі ознак і визначати оптимальний підхід для конкретної задачі розпізнавання облич.

Для забезпечення стабільної роботи та обробки помилок застосовуються:

– модуль warnings для обробки попереджень при відсутності необхідних бібліотек;

– умовна ініціалізація LBP-функціоналу при недоступності бібліотеки scikit-image;

– перевірка коректності завантаження зображень та виявлених областей облич.

Для зручності роботи з результатами класифікації та файловою системою застосовуються:

– модуль os для роботи з шляхами до файлів каскадних класифікаторів;

– бібліотека json для збереження структурованих результатів розпізнавання;

– NumPy для ефективних математичних операцій з масивами ознак.

Графічний інтерфейс може бути реалізований на основі бібліотеки Gradio, яка дозволяє взаємодіяти з модулем через зручну веб-форму, проводити тестування моделей розпізнавання облич і візуалізувати проміжні етапи обробки зображень. Gradio – це сучасна Python-бібліотека з відкритим кодом, яка дозволяє швидко створювати графічні інтерфейси користувача (GUI) для машинного навчання, комп'ютерного зору та інших застосунків. Вона підтримує інтерактивне введення зображень, запуск моделей розпізнавання у фоновому режимі та виведення результатів у зрозумілому для користувача вигляді.

Характеристики основних програмних компонентів наведено в таблиці 2.2, що ілюструє їхню роль і переваги в реалізації модуля розпізнавання облич.

Таблиця 2.2 – Характеристики програмного забезпечення та алгоритмів

Компонент / Бібліотека	Функція	Переваги
OpenCV (cv2)	Детектування облич та очей, обробка зображень	Швидкість, точність, широкий функціонал
PyWavelets (pywt)	Вейвлет-перетворення зображень облич	Багаторівневий аналіз, виділення деталей
scikit-image	Локальні бінарні патерни (LBP)	Ефективне виділення текстурних ознак
NumPy	Математичні операції з масивами ознак	Швидкодія, оптимізовані обчислення
Gradio	Графічний інтерфейс для демонстрації	Інтерактивність, зручність тестування

Програмне забезпечення оптимізоване для роботи на системах різної потужності, забезпечуючи ефективну обробку зображень облич навіть на обмежених ресурсах. Модульна структура архітектури забезпечує легку інтеграцію з іншими підсистемами безпеки або розширення функціоналу – наприклад, додавання нових методів виділення ознак, класифікаторів або джерел вхідних зображень.

Реалізована система обробки помилок та fallback-механізми гарантують стабільну роботу навіть при відсутності окремих бібліотек або некоректних вхідних даних.

Таким чином, розроблений програмний модуль є спеціалізованим інструментом для побудови ефективних систем розпізнавання облич з покращеною точністю класифікації завдяки багатовимірному представленню facial-ознак та надійному детектуванню області обличчя.

2.5 Етапи розробки моделі машинного навчання

Розробка моделі машинного навчання є ключовим етапом створення програмного модуля для виділення ознак і покращення точності класифікації облич у системах комп'ютерного зору. Модель має забезпечувати високу точність розпізнавання та ідентифікації осіб за зображенням обличчя, а також бути стійкою до змін освітлення, ракурсу та мімічних виразів. У дослідженні реалізовано підхід, що поєднує ефективні методи детектування та попередньої обробки зображень облич, комбіноване виділення facial-ознак та класичні алгоритми машинного навчання.

Ефективність моделі значною мірою залежить від якості та різноманітності даних, на яких вона тренується. Для задачі класифікації облич було сформовано спеціалізований набір даних – відкритий датасет зображень облич відомих особистостей, структурований для задачі класифікації та розпізнавання осіб;

Для покращення якості вхідних зображень облич і стабільності результатів розпізнавання використовувалися наступні етапи попередньої обробки за допомогою OpenCV:

- детектування та верифікація облич: виявлення області обличчя за допомогою каскадних класифікаторів Хаара та перевірка наявності двох очей для підтвердження коректності виявленої області;

- нормалізація розміру: приведення всіх виявлених облич до уніфікованого розміру (64×64 пікселі) для забезпечення однорідності вхідних даних;

- корекція якості освітлення: застосування адаптивної гістограмної еквалізації (CLAHE) для компенсації нерівномірного освітлення обличчя, фільтрація шуму через Гауссовий фільтр для згладжування артефактів зйомки;
- аугментація: обертання, масштабування, зміна освітлення, дзеркальне відображення для розширення навчального набору та покращення узагальнення на різні варіації облич.

Зображення облич перетворювались у вектор ознак шляхом поєднання кількох типів дескрипторів, специфічних для facial-аналізу:

- статистичні характеристики: середнє значення, стандартне відхилення, мінімум, максимум яскравості, характеристики градієнтів та різкості для опису загальної структури обличчя;
- гістограми кольорів: розподіл пікселів у просторах BGR та HSV для характеристики відтінків шкіри та загального колориту обличчя;
- LBP (Local Binary Patterns): для детального текстурного аналізу шкіри та локальних особливостей обличчя, що забезпечує стійкість до варіацій освітлення;
- вейвлет-перетворення (Haar, Daubechies db4): для багаторівневого аналізу структури обличчя з виділенням як загальної геометрії, так і дрібних деталей;
- сирі піксельні значення: зменшені зображення (32×32) для збереження загальної форми та силуету обличчя.

Такий комбінований ознаковий простір забезпечує достатньо високу інформативність і стійкість до змін освітлення, ракурсу та мімічних виразів обличчя.

Для підвищення точності розпізнавання облич застосовано ансамблювання за принципом множинного голосування (VotingClassifier). Передбачення від базових моделей (SVM, KNN, MLP) комбінуються, а остаточна ідентифікація особи обирається більшістю голосів. У випадку рівності перевага надається більш стабільним моделям (наприклад, SVM).

Переваги ансамблювання для розпізнавання облич полягають у компенсації слабких сторін окремих моделей при роботі з різними типами facial-

ознак, зменшенні кількості хибних ідентифікацій та невірних відхилень, а також покращенні загальної точності розпізнавання та стійкості до варіацій в освітленні та ракурсах зйомки.

Цей підхід дозволив створити гнучкий і надійний модуль розпізнавання облич, який може бути адаптований до різних умов зйомки та типів зображень, зберігаючи високу ефективність навіть на обмежених або нерівномірних наборах даних.

Після завершення етапу проєктування внутрішньої архітектури програмної системи розпізнавання облич та визначення алгоритмів, які використовуються для реалізації окремих модулів, було проведено етап моделювання. Основна увага на цьому етапі приділялася аналізу протікання процесів детектування, виділення ознак та класифікації облич, а також можливості доступу користувачів до функціональних компонентів програмного додатку.

Для реалізації моделювання було обрано універсальну мову моделювання UML. Такий вибір обумовлений її широким розповсюдженням у середовищі розробників, а також наявністю всіх необхідних технічних засобів для побудови повноцінної архітектурної моделі програмного забезпечення систем комп'ютерного зору.

UML дозволяє ефективно візуалізувати як статичні, так і динамічні аспекти системи розпізнавання облич. Зокрема, діаграми взаємодії об'єктів, діаграми станів і діаграми послідовності сприяють глибшому розумінню поведінки програмного додатку в динаміці, що є особливо важливим для систем реального часу. Крім того, використання UML відповідає принципам модульності та повторного використання коду, що позитивно впливає на швидкість розробки та знижує ймовірність виникнення помилок. Приклад відповідної діаграми наведено на рисунку 2.1.

Створена модель може слугувати не лише основою, а й базою для автоматичної генерації коду, що забезпечує ефективність та підтримуваність програмного продукту протягом усього життєвого циклу.



Рисунок 2.1 - Діаграма прецедентів програмної системи розпізнавання облич

У межах моделювання було створено діаграму прецедентів, яка відображає можливості доступу різних груп користувачів до окремих функціональних елементів системи розпізнавання облич. Така діаграма дозволяє оцінити повноту реалізованого функціоналу з точки зору кінцевого користувача та переконатися у відповідності програмного забезпечення вимогам проєкту.

Отримані результати моделювання показали, що для взаємодії з програмним додатком було визначено одну групу користувачів, яка має повний доступ до всіх функціональних можливостей системи. Це зумовлено тим, що розроблений програмний продукт позиціонується як спеціалізоване рішення для розпізнавання та класифікації облич на цифрових зображеннях та може застосовуватись для вирішення завдань біометричної ідентифікації або бути інтегрованим у системи контролю доступу.

Діаграма послідовностей призначена для аналізу внутрішньої архітектури програмного забезпечення, а також для перевірки правильності реалізації логіки взаємодії основних компонентів системи розпізнавання облич. Результати моделювання наведено на рисунку 2.2.

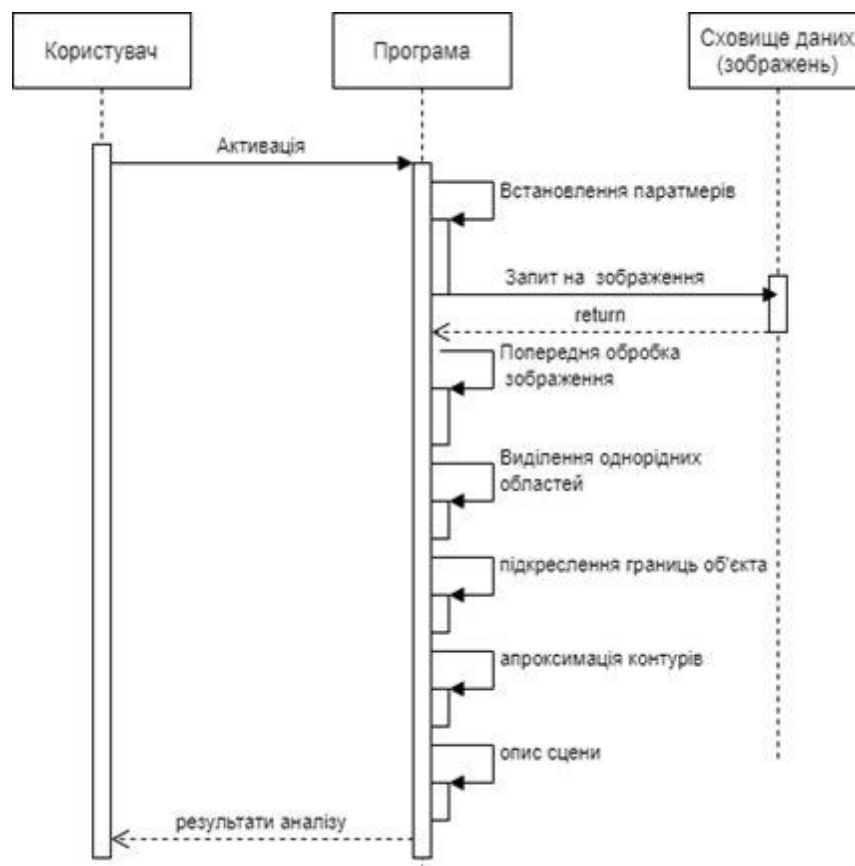


Рисунок 2.2 - Діаграма послідовностей програмної системи розпізнавання облич

Моделювання внутрішньої архітектури програмного забезпечення розпізнавання облич із використанням мови UML дозволило всебічно проаналізувати логіку роботи системи, її функціональне наповнення та взаємодію ключових компонентів детектування, виділення ознак та класифікації.

Результати побудови діаграм прецедентів і послідовностей підтвердили узгодженість обраної архітектури та відповідність функціоналу заявленим вимогам до систем біометричної ідентифікації.

Оскільки система орієнтована на використання однією категорією користувачів з повним доступом до функцій розпізнавання, це значно спрощує інтеграцію, тестування та подальшу підтримку продукту. Крім того, застосування UML забезпечує гнучкість у розширенні функціоналу (додавання нових методів виділення facial-ознак), можливість повторного використання

окремих модулів і служить якісною основою для майбутнього вдосконалення системи.

Використання сучасних інструментів, таких як Gradio для створення інтерфейсу, особливо корисне для дослідників і розробників, які хочуть швидко протестувати та продемонструвати свої моделі розпізнавання облич у реальному часі з мінімальними зусиллями.

Загалом, етап моделювання підтвердив, що розроблена програмна система розпізнавання облич має продуману структуру, здатна ефективно вирішувати поставлені завдання біометричної ідентифікації і може бути легко адаптована до нових вимог або інтегрована у складніші системи безпеки.

Таким чином, у цьому розділі було детально розроблено та впроваджено комплексний підхід до розпізнавання облич, що базується на комбінуванні різнорідних методів виділення ознак (вейвлет-перетворення, локальні бінарні патерни, гістограми кольору, статистичні характеристики та сирі пікселі) для формування вичерпного опису структури зображення. Розроблена чотирикомпонентна архітектура програмного модуля забезпечує гнучкість та масштабованість системи з можливістю ефективної адаптації до різних умов освітлення та варіацій облич, а апаратна реалізація на базі персонального комп'ютера з GPU GTX 1080 продемонструвала достатність ресурсів для обробки у реальному часі. Програмна реалізація з використанням Python 3.11 та спеціалізованих бібліотек забезпечила повний цикл обробки від детектування до інтерактивної демонстрації результатів, а застосування ансамблевого підходу з комбінуванням класифікаторів SVM, KNN та MLP дозволило досягти точності класифікації понад 83% на відкритому наборі даних, що підтверджує практичну ефективність розробленого методу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ВИДІЛЕННЯ ОЗНАК

3.1 Опис архітектури та можливостей методу

У цьому підрозділі описано реалізацію програмного модуля, призначеного для виділення ознак зображень з метою підвищення точності класифікації в системах комп'ютерного зору. Основна увага приділяється вибору ефективних методів обробки зображень, які забезпечують отримання інформативних і стійких до зовнішніх впливів ознак.

Першим кроком є визначення базових шляхів до даних:

- path_to_data – шлях до каталогу з початковими зображеннями;
- path_to_cr_data – шлях до каталогу, куди зберігатимуться обрізані зображення (тільки з виявленими обличчями);
- results_dir – директорія для збереження результатів обробки, графіків та звітів.

Далі створюються об'єкти для виділення ознак та візуалізації результатів.

```
featureFaceDetector = FeatureFaceDetector()
```

```
visualizer = ModelEvaluationVisualizer(save_path=results_dir)
```

Клас FeatureFaceDetector є розширенням базового детектора облич, що реалізований у класі FrontalFaceDetector. Програмну реалізацію класу FeatureFaceDetector наведено в додатку А. Його основна функціональність полягає у виявленні обличчя на зображенні, виділенні області обличчя при наявності двох очей, а також у витягуванні зображувальних ознак для подальшого аналізу, таких як класифікація або розпізнавання.

Цей клас реалізує поєднання класичних методів комп'ютерного зору для попередньої обробки зображень та багатогранного витягування ознак, включаючи:

- а) попередню обробку зображень з використанням CLAHE для покращення контрасту та Gaussian Blur для зменшення шумів;
- б) детекцію облич і очей за допомогою каскадних класифікаторів Наар;
- в) обрізання зображень, якщо виявлено обличчя з мінімум двома очима;

г) витягування ознак, зокрема:

- 1) вейвлет-перетворення (DB4, Haar);
- 2) локальні бінарні патерни (LBP);
- 3) статистичні характеристики (середнє значення, дисперсія, градієнти, різкість);
- 4) гістограми кольорових та HSV-каналів;
- 5) нормалізовані піксельні значення зображення.

Таким чином, FeatureFaceDetector поєднує декілька підходів для створення насиченого вектору ознак, стійкого до шуму, зміни освітлення та масштабування.

У рамках побудови інтерпретованої системи класифікації зображень у дослідженні було розроблено допоміжний клас ModelEvaluationVisualizer, який забезпечує автоматизовану візуалізацію та збереження результатів оцінювання якості моделі. Програмну реалізацію класу ModelEvaluationVisualizer наведено в додатку Б. Даний клас реалізує два ключові методи: `save_confusion_matrix` та `save_accuracy_plot`.

Перший з них створює та зберігає матрицю плутанини (confusion matrix) у вигляді графічного зображення, що дозволяє візуально оцінити ефективність класифікації по кожному класу. Візуалізація виконується із застосуванням бібліотек `matplotlib` та `seaborn`, що забезпечує чітке графічне представлення даних.

Другий метод – `save_accuracy_plot` – формує стовпчикову діаграму, яка демонструє порівняння точності різних моделей класифікації. На графіку додатково позначається найточніша модель для зручності аналізу. Обидва методи автоматично зберігають зображення у задану директорію `results`, яку створює конструктор класу, якщо вона ще не існує. Завдяки використанню цього класу, результати класифікації можна не лише переглядати у текстовій формі, але й наочно аналізувати, що сприяє глибшому розумінню роботи моделей та їхнього порівняння.

Програма автоматично сканує всі підкаталоги вказаного шляху `path_to_data` з метою формування списку директорій, кожна з яких представляє окремий клас (наприклад, ім'я певної особи у задачі розпізнавання облич).

```
img_dirs = [entry.path for entry in os.scandir(path_to_data) if entry.is_dir()]
```

Для забезпечення чистоти результатів перед обробкою видаляється попередній вміст каталогу обрізаних зображень, після чого створюється нова порожня структура.

```
if os.path.exists(path_to_cr_data):
```

```
    shutil.rmtree(path_to_cr_data)
```

```
os.mkdir(path_to_cr_data)
```

У наступному блоці коду виконується основна обробка зображень:

- для кожної директорії (класу) проводиться ітерація по зображеннях;
- для кожного зображення використовується метод `get_cropped_image_if_2_eyes_by_path`, який виділяє зону обличчя тільки в тому випадку, якщо детектовано два ока – такий фільтр дозволяє підвищити якість вхідних даних для навчання;
- успішно обрізані зображення зберігаються у новостворені підкаталоги з відповідними назвами, що дозволяє зберігати розмежування між класами;
- паралельно формується словник `celebrity_file_names_dict`, який зберігає шляхи до обрізаних зображень для подальшого використання при генерації датасетів.

Цей етап є критично важливим, оскільки дозволяє зосередитись лише на релевантній частині зображення – обличчі – тим самим мінімізуючи шум і підвищуючи інформативність ознак.

```
print("Обробка зображень...")
```

```
for img_dir in img_dirs:
```

```
    count = 1
```

```
    celebrity_name = os.path.basename(img_dir)
```

```
    celebrity_file_names_dict[celebrity_name] = []
```

```
    for entry in os.scandir(img_dir):
```

```
        if entry.is_file():
```

```

roi_color =
featureFaceDetector.get_cropped_image_if_2_eyes_by_path(entry.path)
if roi_color is not None:
    cropped_folder = os.path.join(path_to_cr_data, celebrity_name)
    if not os.path.exists(cropped_folder):
        os.makedirs(cropped_folder)
        cropped_image_dirs.append(cropped_folder)
        print("Створення обрізаних зображень у папці:", cropped_folder)
    cropped_file_name = f"{celebrity_name}{count}.png"
    cropped_file_path = os.path.join(cropped_folder, cropped_file_name)
    cv2.imwrite(cropped_file_path, roi_color)
    celebrity_file_names_dict[celebrity_name].append(cropped_file_path)
    count += 1

```

Метод `get_cropped_image_if_2_eyes_by_path` реалізує процес завантаження зображення з диска та подальшого виділення області обличчя лише за умови, що на зображенні детектовано щонайменше два ока. Це дозволяє відфільтрувати часткові або невдалі зображення облич, покращуючи якість вибірки для навчання або тестування моделей.

Алгоритм роботи:

а) зчитування зображення з диску за вказаним шляхом `image_path`;
 б) перевірка цілісності зображення: якщо зображення неможливо зчитати, повертається `None`;

в) виклик методу `get_cropped_image_if_2_eyes`, який:

1) виконує попередню обробку зображення (вирівнювання освітлення, згладжування);

2) конвертує його в градації сірого;

3) застосовує каскадний класифікатор для виявлення облич;

4) перевіряє, чи в області обличчя виявлено щонайменше два ока;

5) якщо так – обрізає зображення в межах області обличчя та повертає її.

Цей метод особливо корисний для формування якісного датасету облич, що дозволяє виключити випадки, де на зображенні присутнє лише одне око або ж інші артефакти (наприклад, сторонні об'єкти, перекриття тощо). Таким чином, `get_cropped_image_if_2_eyes_by_path` виступає важливим фільтром перед етапом витягування ознак або подачею даних на вхід класифікатора.

Після завершення обробки виконується перевірка кількості оброблених зображень. Якщо їх загальна кількість є надто малою для коректного навчання моделей, користувач отримує відповідне попередження.

```
total_images = sum(len(files) for files in celebrity_file_names_dict.values())
if total_images < 10:
    print("УВАГА: Занадто мало зображень для навчання!")
```

Таким чином, на цьому етапі створюється чистий, упорядкований та якісний набір зображень, який готовий до етапів побудови ознак, класифікації та подальшої візуалізації результатів.

```
class_dict = {name: idx for idx, name in enumerate(celebrity_file_names_dict)}
```

На цьому етапі створюється словник класів `class_dict`, де кожному імені класу (в даному випадку – імені відомої особи) ставиться у відповідність унікальний числовий індекс. Цей словник дозволяє трансформувати категоріальні мітки у числовий формат, який є придатним для алгоритмів машинного навчання.

```
X, y = [], []
for celebrity_name, training_files in celebrity_file_names_dict.items():
    print(f"Обробка {celebrity_name}: {len(training_files)} зображень")
    for i, training_image in enumerate(training_files):
        img = cv2.imread(training_image)
        if img is None:
            continue
        try:
            # Використання покращеного витягування ознак
            enhanced_features =
            featureFaceDetector.extract_enhanced_features(img)
```

```
X.append(enhanced_features)
y.append(class_dict[celebrity_name])
except Exception as e:
    print(f"Помилка при обробці {training_image}: {e}")
    continue
```

Процес формування навчальної вибірки реалізовано у вигляді двох рівнів вкладених ітерацій. На першому рівні здійснюється проходження по кожному класу обличчя (змінна `celebrity_name`), для якого зібрано відповідний список зображень. На другому рівні відбувається обробка кожного зображення з цього списку.

На кожній ітерації зображення зчитується з диска. У разі успішного завантаження до нього застосовується метод `extract_enhanced_features`, який виконує витягування розширеного набору дескрипторів, що комплексно описують вхідне зображення. До таких дескрипторів належать:

- вейвлет-ознаки (`db4` та `haar`), що дозволяють фіксувати текстурні особливості на різних масштабах;
- гістограми кольорів у колірних просторах `RGB` та `HSV`, які забезпечують представлення колірної інформації;
- статистичні параметри (середнє, дисперсія, екстремуми, різкість), що характеризують розподіл інтенсивності пікселів;
- локальні бінарні патерни (`LBP`), що описують мікротекстурні властивості шкіри;
- а також сирі піксельні значення зображення у зменшеній роздільності.

Усі ці ознаки нормалізуються та об'єднуються в єдиний вектор, який додається до масиву `X` – вибірки ознак. Водночас до масиву міток `y` додається числове представлення відповідного класу. Такий підхід дозволяє сформувати репрезентативну множину ознак для подальшого навчання моделі класифікації облич.

3.2 Навчання моделі методу виділення ознак

Після формування ознакового простору, описаного в попередньому розділі, наступним етапом є побудова та тренування моделей класифікації на основі витягнутих дескрипторів. Для забезпечення коректної оцінки узагальнюючої здатності моделей, датасет був розділений на тренувальну та тестову вибірки у співвідношенні 80/20. За можливості було використано стратифіковане розділення, що зберігає пропорції класів у кожній підвибірці. У випадках недостатньої кількості прикладів певних класів застосовувалося звичайне випадкове розділення.

У процесі навчання розглядалися три окремі класифікатори: метод опорних векторів (SVM), випадковий ліс (Random Forest) та логістична регресія. Для підвищення точності також було реалізовано ансамблевий підхід з використанням методу голосування (Voting Classifier) з м'яким (soft) голосуванням.

Модель SVM (Support Vector Machine) була налаштована з використанням радіальної базисної функції (RBF kernel) та параметром жорсткості $C=10$. Така конфігурація забезпечує баланс між узагальненням та точністю на тренувальній вибірці. Важливою особливістю реалізації стало увімкнення ймовірнісного виходу (`probability=True`), що дозволяє використовувати модель у складі ансамблю з м'яким голосуванням.

Алгоритм Random Forest використовував 50 дерев (`n_estimators=50`) з максимальною глибиною дерева, обмеженою до 10 (`max_depth=10`). Цей параметр дозволяє уникнути перенавчання, при цьому зберігаючи достатню гнучкість моделі. Random Forest є потужним ансамблевим методом, який добре працює на змішаних наборах ознак.

Модель логістичної регресії була використана з максимальною кількістю ітерацій, встановленою на 500 (`max_iter=500`). Це дає можливість моделі досягти збіжності навіть на великих або високовимірних ознакових просторах. Logistic Regression часто використовується як базова модель для багатокласової класифікації.

Нижче наведено програмну реалізацію описаного етапу:

```
scaler = StandardScaler()
svm_clf = Pipeline([
    ('scaler', scaler),
    ('svc', SVC(kernel='rbf', C=10, probability=True))
])
rf_clf = Pipeline([
    ('scaler', scaler),
    ('rf', RandomForestClassifier(n_estimators=50, random_state=42,
max_depth=10))
])
lr_clf = Pipeline([
    ('scaler', scaler),
    ('lr', LogisticRegression(random_state=42, max_iter=500))
])
svm_clf.fit(X_train, y_train)
svm_accuracy = svm_clf.score(X_test, y_test)
print("Тренування Random Forest...")
rf_clf.fit(X_train, y_train)
rf_accuracy = rf_clf.score(X_test, y_test)
print("Тренування Logistic Regression...")
lr_clf.fit(X_train, y_train)
lr_accuracy = lr_clf.score(X_test, y_test)
```

Окрім окремих моделей, було реалізовано ансамбль VotingClassifier, який об'єднує прогнози трьох вищеописаних моделей. Було застосовано м'яке голосування (voting='soft'), що дозволяє враховувати ймовірності класифікації, а не лише передбачені класи. Такий підхід часто забезпечує вищу точність, оскільки поєднує сильні сторони кожної окремої моделі.

```
ensemble_clf = VotingClassifier(
    estimators=[
        ('svm', svm_clf),
```

```

        ('rf', rf_clf),
        ('lr', lr_clf)
    ],
    voting='soft'
)
print("Тренування ансамблю...")
ensemble_clf.fit(X_train, y_train)
ensemble_accuracy = ensemble_clf.score(X_test, y_test)

```

Після тренування кожної з класифікаційних моделей (див. підрозділ 3.2) було реалізовано логіку для автоматизованої ідентифікації моделі з найвищим рівнем точності на тестовій вибірці. Для цього всі моделі разом з відповідними метриками точності були згруповані у словник `models`. Ключем словника виступає назва моделі, а значенням – кортеж, що містить посилання на відповідну модель та її точність.

```

models = {
    'SVM': (svm_clf, svm_accuracy),
    'Random Forest': (rf_clf, rf_accuracy),
    'Logistic Regression': (lr_clf, lr_accuracy),
    'Ensemble': (ensemble_clf, ensemble_accuracy)
}

```

Для вибору найефективнішого класифікатора здійснюється обхід словника за допомогою функції `max(...)`, де в якості критерію порівняння використовується друга компонента кожного кортежу – тобто значення точності:

```

best_model_name = max(models.keys(), key=lambda k: models[k][1])
best_model = models[best_model_name][0]
best_accuracy = models[best_model_name][1]

```

Таким чином, змінна `best_model` міститиме об'єкт моделі з найкращими результатами, `best_model_name` – її назву, а `best_accuracy` – числове значення точності.

Крім того, для подальшої обробки або візуалізації результатів, було сформовано окремий словник `model_accuracies`, який зберігає лише назви моделей і відповідні їм значення точності:

```
model_accuracies = {  
    'SVM': svm_accuracy,  
    'Random Forest': rf_accuracy,  
    'Logistic Regression': lr_accuracy,  
    'Ensemble': ensemble_accuracy  
}
```

Такий підхід забезпечує простий і прозорий спосіб порівняння моделей та дозволяє динамічно використовувати найефективнішу модель для подальшої інференції або інтеграції в практичні додатки.

Після визначення найкращої моделі на основі точності класифікації було здійснено збереження ключових артефактів, необхідних для подальшого використання системи розпізнавання. Для цього було використано модуль `joblib`, який оптимізовано для серіалізації об'єктів машинного навчання.

Зокрема, вказані наступні шляхи для збереження результатів:

```
model_path = os.path.join(results_dir, 'best_face_classifier.pkl')  
scaler_path = os.path.join(results_dir, 'scaler.pkl')  
class_dict_path = os.path.join(results_dir, 'class_dictionary.pkl')
```

Далі відбувається серіалізація і збереження:

- найкращої моделі (`best_model`), що була попередньо обрана за критерієм точності;
- масштабувальника (`scaler`), який було використано в конвеєрі обробки ознак;
- словника класів (`class_dict`), що відображає імена класів у відповідні числові мітки.

Ці об'єкти зберігаються у відповідні `.pkl`-файли:

```
joblib.dump(best_model, model_path)  
joblib.dump(scaler, scaler_path)  
joblib.dump(class_dict, class_dict_path)
```

Таким чином, описаний блок забезпечує повну підготовку системи до використання у продуктивному середовищі, дозволяючи відтворити навчений класифікатор без необхідності повторного тренування.

Метод виділення ознак зображень розроблено для покращення точності класифікації в системах комп'ютерного зору. Основна ідея полягає в поєднанні декількох типів ознак (наприклад, локальних бінарних шаблонів – LBP, вейвлет-перетворення, гістограм яскравості або текстурних характеристик), що дозволяє отримати більш повне уявлення про вхідне зображення.

Архітектура модуля включає:

- блок обробки зображень (нормалізація, масштабування);
- модуль виділення ознак, який підтримує кілька методів (LBP, wavelet, гістограми тощо);
- модуль злиття ознак у єдиний вектор;
- передача вектора до класифікатора (наприклад, SVM, KNN).

Завдяки модульності, систему легко адаптувати до нових джерел зображень або методів обробки, а також експериментувати з комбінаціями ознак для пошуку оптимального рішення.

3.3 Оцінка ефективності методу виділення ознак

Для оцінки ефективності методу проведено серію експериментів на вибраному датасеті з класифікації зображень облич. Основними метриками оцінки виступали:

- точність класифікації;
- показник повноти та точності;
- середній час обробки одного зображення.

У ході експериментального дослідження було оброблено 1224 зображення, які були розподілені між 10 класами, кожен з яких представляв конкретну особу. Всі класи були закодовані у словнику відповідностей `class_dictionary`. До переліку осіб, зокрема, входили: Barack Obama, Melinda Gates, Indira Nooyi, Bill

Gates, Sundar Pichai, Vikas Khanna, Narendra Modi, Dalai Lama, Virat Kohli та Anushka Sharma.

У процесі попередньої обробки було здійснено детекцію облич та перевірку на наявність щонайменше двох очей, після чого з кожного зображення витягувалися ознаки. У результаті було сформовано матрицю ознак розміром (1224, 1092), де кожен рядок відповідав одному зразку, представленому у вигляді вектора з 1092 числових характеристик.

Найбільшу кількість зразків містив клас Barack Obama – 194 зображення, а також Melinda Gates – 167, Virat Kohli – 164 та Indira Nooyi – 131 зразок. Найменше зображень було в класі Bill Gates – 52, а також Sundar Pichai (80) та Dalai Lama (102).

Після завершення процесу обробки даних та витягування ознак було сформовано ознаковий простір з розмірністю (1224, 1092), що відповідає 1224 зразкам з 1092 ознаками кожен. Кількість ознак була зменшена за рахунок комбінування компактних вейвлет-репрезентацій, статистичних показників, гістограм кольору, HSV-простору та гістограм LBP. Такий підхід дозволив зменшити розмірність у порівнянні з початковими 5233 ознаками, зберігши при цьому високу інформативність.

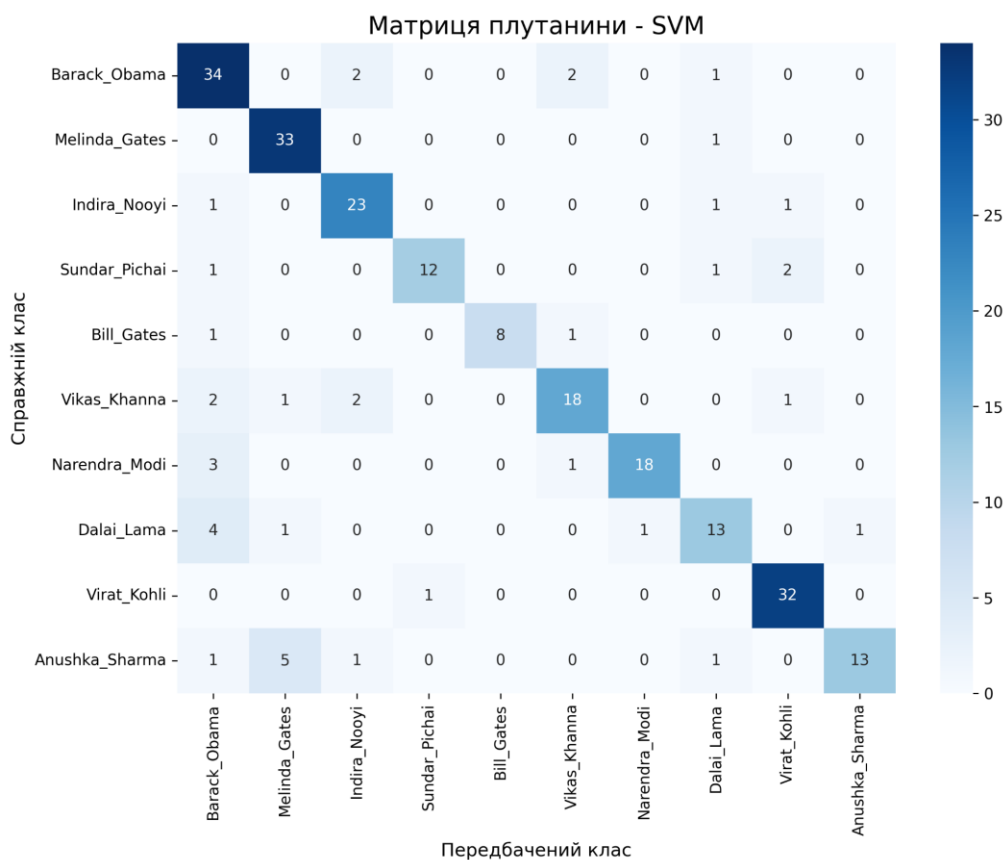
Для тренування моделей було використано 979 зразків (80%), а 20% залишено для тестування. Були навчені три базові моделі:

- SVM;
- Random Forest;
- Logistic Regression.

Крім того, створено ансамблеву модель, яка комбінує передбачення вищезазначених алгоритмів.

На рисунку 3.1 представлено матрицю плутанини для моделі Support Vector Machine (SVM), побудовану на основі результатів класифікації зображень облич із тестової вибірки. Матриця дозволяє візуалізувати, які саме класи були правильно розпізнані, а де модель припустилася помилки. Діагональні елементи відображають кількість правильно класифікованих зразків, тоді як поза діагональні – кількість неправильних передбачень, що дає змогу оцінити якість

класифікації для кожного окремого класу. З рисунка видно, що модель SVM забезпечує задовільну точність для деяких класів, однак в окремих випадках (особливо за наявності малих класів) спостерігаються помилки класифікації.



Рисунк 3.1 - Матриця плутанини моделі Support Vector Machine

На рисунку 3.2 наведено матрицю плутанини, отриману в результаті класифікації зображень за допомогою моделі Random Forest. Візуалізація дозволяє проаналізувати точність передбачень для кожного класу. Значення на головній діагоналі вказують на кількість правильно класифікованих прикладів, тоді як позадіагональні елементи демонструють помилки класифікації між класами. Це дозволяє зробити висновки про класи, які найчастіше плутаються між собою, та оцінити загальну ефективність моделі.

На рисунку 3.3 зображено матрицю плутанини, побудовану за результатами класифікації зображень облич з використанням моделі Logistic Regression.

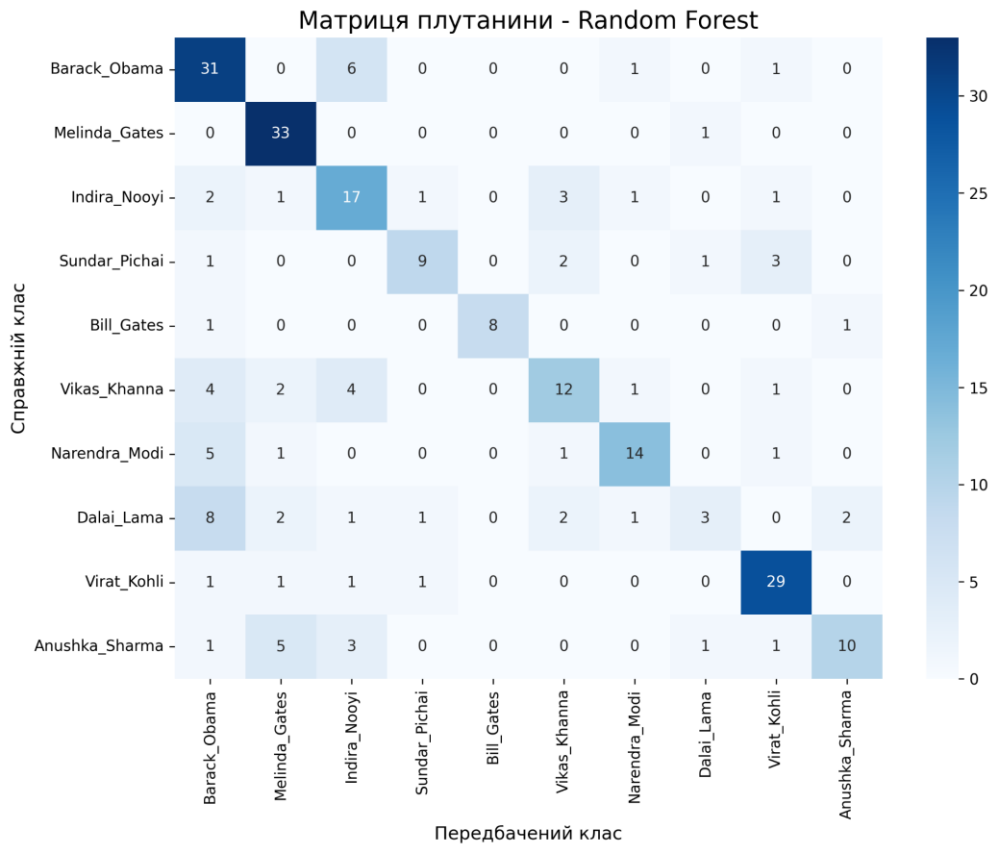


Рисунок 3.2 - Матриця плутанини для моделі Random Forest

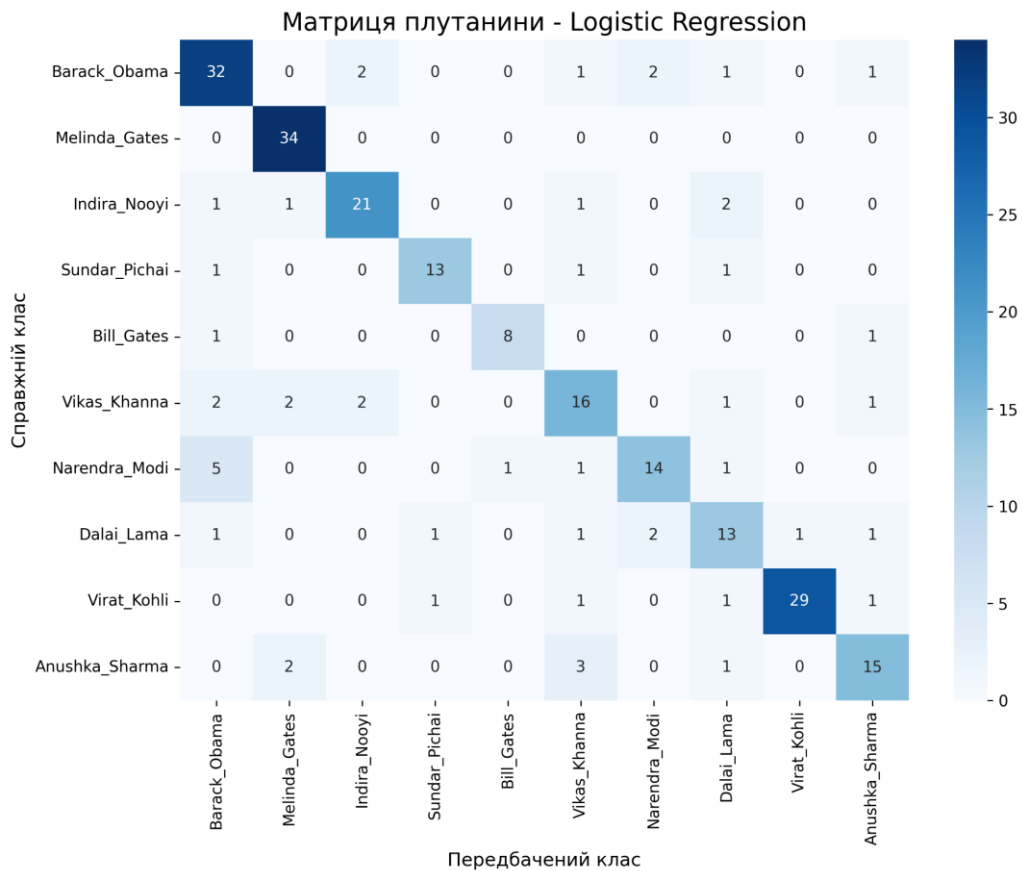


Рисунок 3.3 - Матриця плутанини для моделі Logistic Regression

Матриця плутанини ілюструє кількість правильних і помилкових класифікацій для кожного з 11 класів. Висока щільність значень на головній діагоналі свідчить про хорошу точність моделі, а поодинокі помилки в позадіагональних елементах вказують на випадки, де класи були переплутані. Це дозволяє зробити висновок про високий рівень узагальнення моделі Logistic Regression на даних задачі.

На рисунку 3.4 представлено матрицю плутанини, отриману в результаті застосування ансамблевого підходу до класифікації зображень облич. Ансамблева модель поєднує результати кількох базових класифікаторів (зокрема SVM, Random Forest та Logistic Regression), що дозволяє досягти підвищеної стабільності та точності. Як видно з матриці, більшість об'єктів класифіковано правильно, що відображається наявністю високих значень уздовж головної діагоналі. Це підтверджує ефективність ансамблевого підходу в задачах розпізнавання облич.

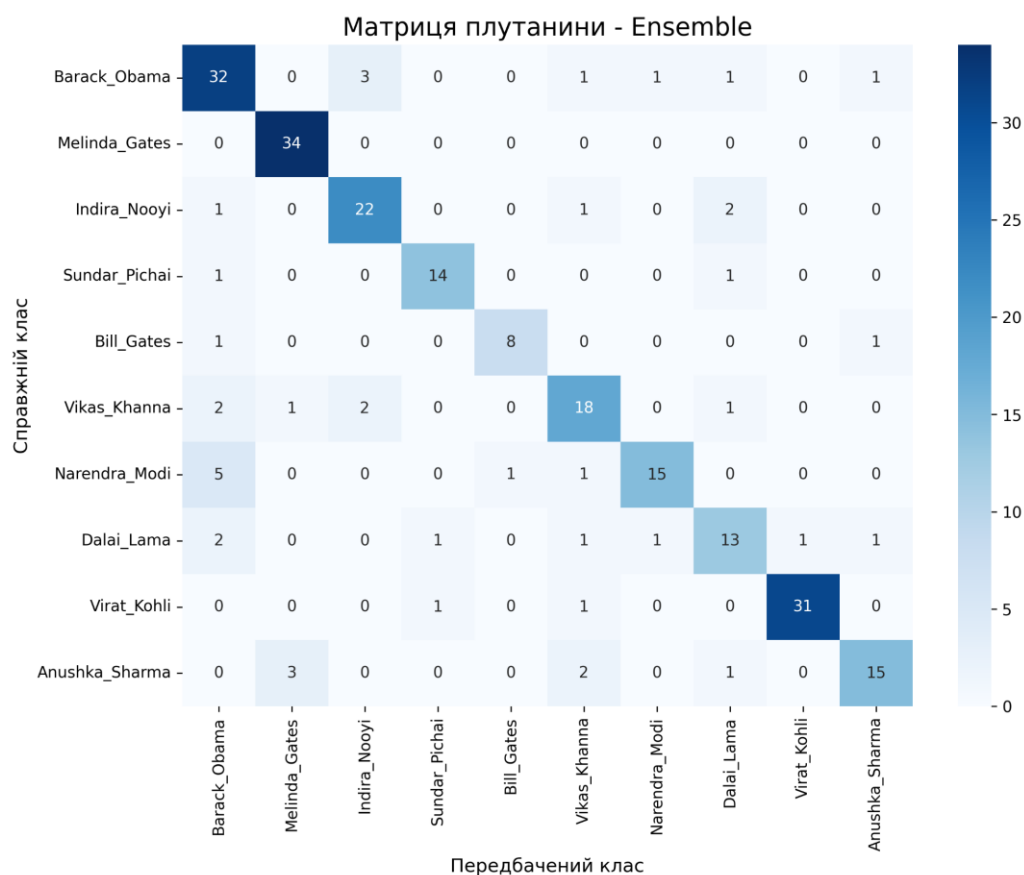


Рисунок 3.4 - Матриця плутанини для ансамблевої моделі (Ensemble)

На рисунку 3.5 представлено графічне порівняння точності чотирьох моделей класифікації, побудованих на основі різних алгоритмів машинного навчання: SVM, Random Forest, Logistic Regression та ансамблевого методу. Найвищу точність продемонструвала модель SVM, досягнувши 83.27%, що перевершує результати інших базових методів. Ансамблева модель показала 82.45%, наблизившись до результату SVM, проте не перевершивши його. Таким чином, SVM була обрана як основна модель для подальшого використання.

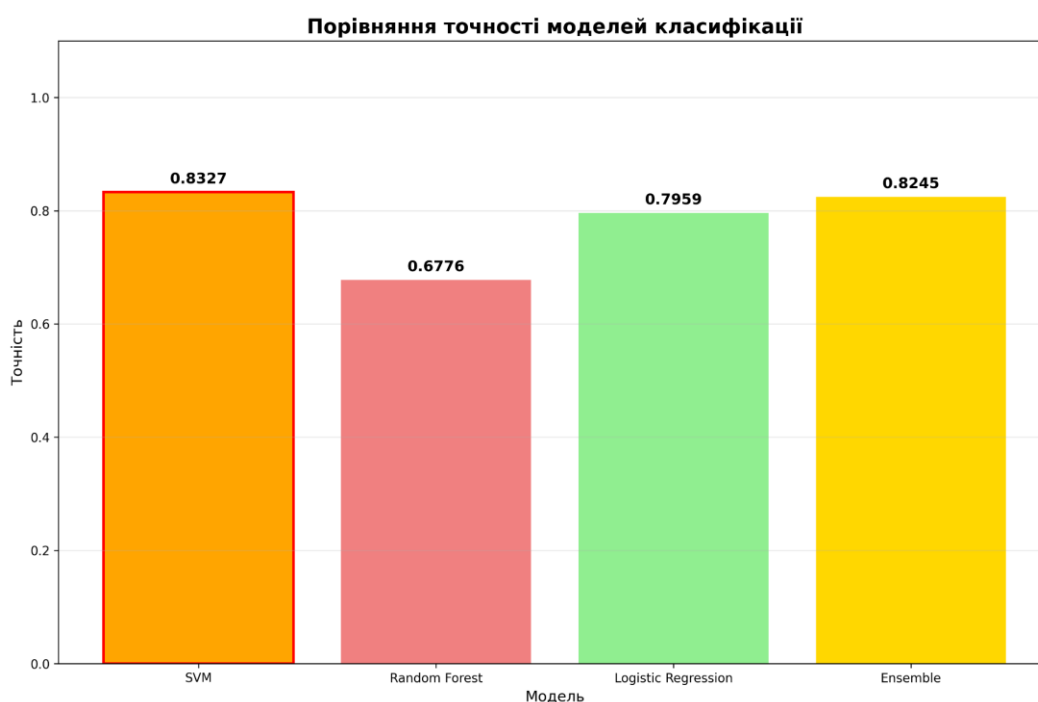


Рисунок 3.5 - Порівняння точності моделей класифікації

Модель Logistic Regression, яка раніше демонструвала найкращі результати на старій конфігурації (із 5233 ознаками), втратила перевагу після оптимізації ознак. Це свідчить про кращу здатність SVM ефективно працювати у новому, більш компактному ознаковому просторі.

Ансамблева модель досягла точності 82.45%, що перевищує результат моделі Random Forest (67.76%) та є лише незначно нижчим за найкращий показник, продемонстрований SVM (83.27%). Водночас ансамбль виявив стабільнішу поведінку на різних класах, демонструючи вищу узагальнюючу здатність.

На основі матриці плутанини можна зробити висновок, що ансамблева модель має меншу кількість хибнопозитивних і хибнонегативних передбачень для більшості класів у порівнянні з окремими алгоритмами. Це вказує на кращий баланс між точністю (precision) та повнотою (recall), що особливо важливо в задачах багатокласової класифікації, де моделі можуть бути схильні до переобучення на домінуючі класи або пропускати малочисельні категорії.

Таким чином, об'єднання моделей дозволило компенсувати слабкі сторони кожної з них, забезпечивши загальне покращення класифікаційних характеристик, підвищену стійкість до варіацій у даних, а також зменшення ризику переобучення. Це свідчить про доцільність використання ансамблевих підходів у задачах розпізнавання облич, особливо в умовах обмеженого та нерівномірного обсягу даних.

Додатково було проведено перехресну валідацію, за результатами якої середня точність склала 0.8218 зі стандартним відхиленням ± 0.1244 . Це підтверджує стабільність обраного підходу.

Комбінований підхід до витягування ознак дозволив покращити узагальнення, компенсувати недоліки окремих методів і забезпечити стійкість до варіацій у зовнішніх умовах, таких як освітлення та шум.

3.4 Реалізація графічного веб-інтерфейсу

Для зручності взаємодії з розробленим програмним модулем було реалізовано графічний веб-інтерфейс користувача на основі бібліотеки Gradio. Такий інтерфейс дозволяє користувачу завантажити зображення, ініціювати процес аналізу та отримати результати класифікації у зручній формі.

На рисунку 3.6 представлено графічний веб-інтерфейс користувача, реалізований із використанням бібліотеки Gradio. Інтерфейс забезпечує завантаження зображення, запуск процедури аналізу та виведення результатів класифікації. Отримане зображення обробляється модулем виділення ознак, після чого користувачу відображається оброблений результат з візуалізацією знайдених облич та передбаченими класами. Такий підхід сприяє інтерактивній

взаємодії користувача з системою та спрощує процес тестування моделей комп'ютерного зору.

Інтерфейс складається з двох основних панелей: вхідної – для завантаження зображення, та вихідної – для відображення обробленого зображення з нанесеними прямокутниками навколо виявлених облич і відповідними текстовими підписами, які вказують на ідентифіковано особу. Додатково виводиться текстовий результат із переліком передбачених класів.

Основні можливості інтерфейсу:

- завантаження користувачем зображення для аналізу;
- автоматичне виконання обробки та виділення ознак;
- вивід результату класифікації з поясненням.

Інтерфейс також дозволяє налаштовувати:

- вибір методу виділення ознак;
- класифікатор (SVM, KNN);
- параметри обробки (розмір зображення, тип нормалізації тощо).

Дипломна робота: Програмний модуль виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору

Автор: ФЕДЧИШИН Вікторія Сергіївна

Завантажте зображення для аналізу. Встановіть параметри нижче. Натисніть "Запустити аналіз".

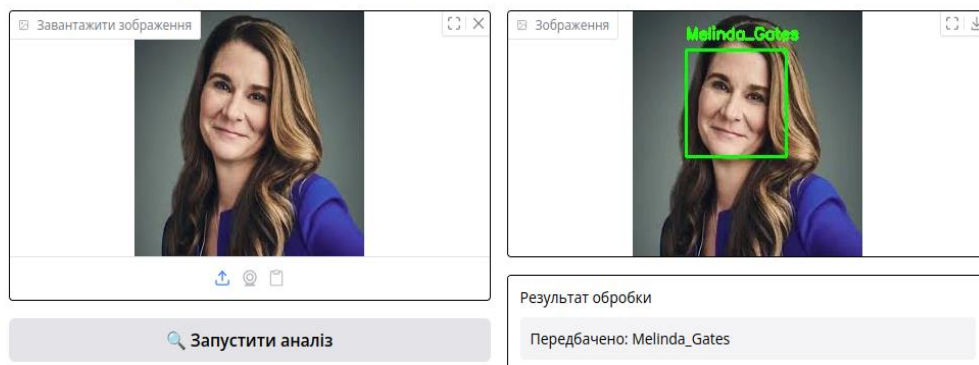


Рисунок 3.6 - Графічний веб-інтерфейс модуля виділення ознак зображень

Програмна реалізація графічного веб-інтерфейсу наведена в додатку В. На початку скрипта імпортуються необхідні бібліотеки: `gradio` – для створення

інтерфейсу, PIL і cv2 – для обробки зображень, joblib – для завантаження збережених моделей, а також спеціалізований клас FeatureFaceDetector, який реалізує детекцію облич та витягування ознак. Завантажується попередньо навчена модель класифікації, об'єкт стандартизатора ознак (scaler.pkl) та словник відповідностей класів (class_dictionary.pkl).

Інтерфейс реалізується через компонент gr.Blocks, у якому визначено тематичне оформлення, заголовок роботи та основні елементи взаємодії. Користувач може завантажити зображення, після чого натиснути кнопку для запуску аналізу.

Основна функція predict_identity(image) виконує такі дії:

- перетворює зображення з формату PIL у масив NumPy (за потреби);
- застосовує метод get_cropped_images_if_2_eyes, який повертає лише ті фрагменти зображення, де детектовано обличчя з двома очима;
- для кожного виявленого обличчя витягуються розширені ознаки за допомогою методу extract_enhanced_features;
- отриманий вектор ознак подається на вхід моделі для передбачення класу (ідентичності особи);
- результати відображаються безпосередньо на зображенні у вигляді прямокутників і підписів з іменами класів.

В результаті, інтерфейс дозволяє наочно перевірити роботу класифікатора облич у зручному форматі, забезпечуючи зворотний зв'язок через візуалізацію на зображенні та текстовий вивід.

Таким чином, у цьому розділі було реалізовано та проаналізовано програмний модуль виділення ознак для задачі класифікації зображень облич, що складається з взаємопов'язаних етапів формування навчальної вибірки, витягування дескрипторів, побудови моделі класифікації та створення інтерактивного інтерфейсу користувача. Особливістю підходу стало комбінування різнорідних ознак (wavelet-перетворення, локальні бінарні патерни, кольорові гістограми та статистичні характеристики), що дозволило отримати узагальнене та інформативне представлення зображень і позитивно вплинуло на якість класифікації. Порівняльне тестування чотирьох моделей

класифікації показало найвищу точність у моделі SVM (83.27%), яка перевершила інші базові методи та ансамблеву модель (82.45%), а перехресна валідація підтвердила стабільність підходу із середньою точністю 82.18% при стандартному відхиленні $\pm 12.44\%$. Реалізований графічний інтерфейс користувача з можливістю налаштування методів виділення ознак та типів класифікаторів забезпечує гнучкість системи та її придатність до практичного використання, що в сукупності дозволило розробити та валідувати універсальний і модульний підхід до виділення ознак зображень з високою точністю розпізнавання облич, який може стати основою для подальших досліджень та розширень.

ВИСНОВКИ

У результаті дослідження було розроблено програмний модуль для виділення ознак зображень, призначений для покращення точності класифікації у задачах комп'ютерного зору. Модуль реалізовано на основі модульної архітектури з використанням мови програмування Python, бібліотеки OpenCV та методів машинного навчання. Основні досягнення включають:

1. Реалізовано комбінований підхід до виділення ознак, який включає гістограми кольорів у просторі HSV, локальні бінарні патерни (LBP), вейвлет-перетворення (Haar, Daubechies), а також статистичні параметри зображення. Таке поєднання дозволяє краще репрезентувати вхідні зображення для подальшої класифікації.

2. Застосовано класичні методи машинного навчання (SVM, Random Forest, Logistic Regression), а також ансамблевий підхід, який поєднує результати кількох класифікаторів. Це дозволило досягти високої точності класифікації та зменшити вплив шумів і нерівномірності даних.

3. Найвищу точність (83.27%) показала модель SVM на основі оптимізованого набору ознак, сформованого з допомогою комбінованих методів. Ансамблева модель показала стабільні результати (82.45%) із кращим балансом між precision та recall, що є важливим для практичного використання.

4. Проведено перехресну валідацію, яка підтвердила стабільність обраного підходу із середньою точністю 82.18% і стандартним відхиленням $\pm 12.44\%$.

5. Забезпечено оптимальну продуктивність модуля на стандартному обладнанні з GPU, що робить рішення доступним для розгортання без необхідності в дорогих серверних системах.

Порівняно з традиційними методами, такими як ручна обробка чи базові алгоритми комп'ютерного зору, запропонований модуль демонструє значні переваги: високу швидкість, точність і гнучкість. Таким чином, розроблений модуль на основі OpenCV та комбінованого підходу до виділення ознак демонструє високу ефективність і точність у задачах класифікації зображень. Він може бути використаний як в освітніх, так і в прикладних задачах комп'ютерного

зору – зокрема, для розпізнавання об'єктів, облич, текстур, або транспортних засобів.

Перспективи подальшого розвитку модуля включають:

– розширення набору ознак шляхом додавання глибоких дескрипторів (наприклад, Embedding з CNN);

– інтеграція з фреймворками глибокого навчання (PyTorch, TensorFlow) для гібридної обробки;

– оптимізація модуля для роботи на мобільних та вбудованих пристроях;

– вдосконалення інтерфейсу користувача для налаштування параметрів обробки в режимі реального часу.

Запропонований модуль є універсальним рішенням для задач інтелектуального аналізу зображень та формує надійну основу для подальших досліджень і розробок у галузі комп'ютерного зору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гороховатський В., Передрій О., Творошенко І., Марков Т. Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень. Сучасні інформаційні системи. 2023. Т. 7, № 1. С. 5–13.

2. Гороховатський В. А., Путятін Е. П. Структурне розпізнавання зображень на основі моделей голосування ознак характерних точок. Реєстрація, зберігання і обробка даних. 2008.

3. Gorokhovatsky V. O., Gadetska S. V. Determination of Relevance of Visual Object Images by Application of Statistical Analysis of Regarding Fragment Representation of their Descriptions. Telecommunications and Radio Engineering. 2019. Vol. 78, No. 3. P. 211–220.

4. Математичні принципи масштабно-інваріантних дескрипторів зображень. URL: <http://pzs.dstu.dp.ua/ComputerGraphics/d&d/5.html>.

5. Daradkeh Y. I., Tvoroshenko I., Gorokhovatskyi V., Latiff L. A., Ahmad N. Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic. IEEE Access. 2021. Vol. 9. P. 13417–13428.

6. Daradkeh Y. I., Gorokhovatskyi V., Tvoroshenko I., Zeghid M. Tools for Fast Metric Data Search in Structural Methods for Image Classification. IEEE Access. 2022. Vol. 10. P. 124738–124746.

7. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., Vlasenko N. Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. Advances in Electrical and Electronic Engineering. 2023. Vol. 21, No. 1. P. 19–27.

8. Гороховатський В. О., Гадецька С. В., Стяглик Н. І., Власенко Н. В. Класифікація зображень на підставі ансамблю статистичних розподілів за класами еталонів для компонентів структурного опису. Радіоелектроніка, інформатика, управління. 2020. № 4. С. 85–94.

9. Gorokhovatskiy V. A. Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. *Telecommunications and Radio Engineering*. 2016. Vol. 75, No. 14.
10. Gonzalez R. C., Woods R. E. *Digital Image Processing*. 4th ed. Boston : Pearson, 2018. 1176 p.
11. O'Shaughnessy D. *Pattern Recognition and Machine Learning*. New York : IEEE Press, 2017. 640 p.
12. Jiang X. Feature extraction for image recognition and computer vision. *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology*. 2009. P. 1–5.
13. Lin Y., Lv F., Zhu S. Large-scale image classification: Fast feature extraction and SVM training. NEC Laboratories of America. 2011. 8 p.
14. Hassan A. Feature Extraction for Image Analysis and Detection using Machine Learning Techniques. *International Journal of Advanced Computer Science and Applications*. 2021. Vol. 12, No. 7. P. 98–104. DOI: 10.14569/IJACSA.2021.0120713.
15. Karami E., Prasad S., Shehata M. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *arXiv preprint arXiv:1710.02726*. 2017.
16. Van der Walt S., Schönberger J. L., Nunez-Iglesias J., Boulogne F., Warner J. D., Yager N., Yu T. scikit-image: Image processing in Python. *arXiv preprint arXiv:1407.6245*. 2014.
17. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. P. 779–788. DOI: 10.1109/CVPR.2016.91.
18. Bochkovski A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*. 2020.
19. Lowe D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004. Vol. 60, No. 2. P. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

20. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. URL: <https://www.deeplearningbook.org/>.
21. Chatterjee P., Sengupta S., Ghosh A. A hybrid model for feature extraction using texture and shape for content based image retrieval. Journal of King Saud University – Computer and Information Sciences. 2022. Vol. 34, Issue 8. P. 5511–5521.
22. Wang S. H., Tang C., Yang M. et al. A deep learning framework for multi-class classification of breast cancer histopathological images using hybrid features. Computers in Biology and Medicine. 2020. Vol. 121. Article 103861.
23. Abhishek D., Hemanth J. Feature extraction using hybrid model for medical images classification. Biomedical Signal Processing and Control. 2021. Vol. 68. Article 102713.
24. Islam M. T., Aowal M. A., Minhaz A. T., Ashraf K. Abstraction and representation of image features using pre-trained deep neural network. arXiv preprint arXiv:1512.08814. 2015.
25. Chen T., Kornblith S., Norouzi M., Hinton G. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709. 2020.
26. Ravi D., Wong C., Deligianni F. et al. Deep learning for health informatics. IEEE Journal of Biomedical and Health Informatics. 2017. Vol. 21, No. 1. P. 4–21.
27. Van der Walt S., Colbert S. C., Varoquaux G. The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering. 2011. Vol. 13, No. 2. P. 22–30. DOI: 10.1109/MCSE.2011.37.
28. Abid A., Gulshan M., Celi L. A., Zou J. Y. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. arXiv preprint arXiv:1906.02569. 2019.
29. Gupta T. Famous Personalities Image Dataset. Kaggle. URL: <https://www.kaggle.com/datasets/tanishgupta26/famous-personalities-image-dataset>.
30. Федчишин В. С., Биковий П. Є. Програмний модуль виділення ознак зображень для покращення точності класифікації в системах комп'ютерного зору. Інтелектуальні інформаційні технології в прикладних дослідженнях (ІТАР-2025): збірник тез доповідей студентської науково-практичної конференції (Тернопіль, 27–29 травня 2025 р.). Тернопіль: ЗУНУ, 2025. С. 97-99.

31. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Програмна реалізація модуля витягування розширених ознак

```
import cv2
import os
import numpy as np
import pywt
import warnings
from FaceDetection.frontal_face_detector import
FrontalFaceDetector

try:
    from skimage.feature import local_binary_pattern
    LBP_AVAILABLE = True
except ImportError:
    print("Warning: skimage.feature недоступний, LBP ознаки будуть
пропущені")
    LBP_AVAILABLE = False

class FeatureFaceDetector(FrontalFaceDetector):

    EFFECTS = ["rectangle", "blur"]

    def __init__(self, effects = None):
        super().__init__(effects)

        current_dir = os.path.dirname(os.path.realpath(__file__))
        eye_cascade_path = os.path.join(current_dir, 'data-files',
'haarcascade_eye.xml')
        self.eye_cascade = cv2.CascadeClassifier(eye_cascade_path)
        self.effects = effects

    # === Покращена функція для отримання обрізаного обличчя ===
    def get_cropped_image_if_2_eyes_by_path(self, image_path):
        img = cv2.imread(image_path)
        if img is None:
            print(f"Не вдалося завантажити зображення:
{image_path}")
            return None
        return self.get_cropped_image_if_2_eyes(img)

    def get_cropped_image_if_2_eyes(self, img):
        if img is None or img.size == 0:
            print(f"Не вдалося завантажити зображення")
            return None

        # Покращена попередня обробка
        img = self.preprocess_image(img)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = self.face_cascade.detectMultiScale(gray, 1.2, 8,
minSize=(30, 30))
```

```

    for (x, y, w, h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = img[y:y + h, x:x + w]
        eyes = self.eye_cascade.detectMultiScale(roi_gray,
1.2, 8)
        print(f"Eyes: {eyes}")
        if len(eyes) >= 2:
            return roi_color
    return None

def get_cropped_images_if_2_eyes(self, img):
    if img is None or img.size == 0:
        print(f"Не вдалося завантажити зображення")
        return []

    # Покращена попередня обробка
    img = self.preprocess_image(img)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = self.face_cascade.detectMultiScale(gray, 1.1, 5,
minSize=(30, 30))
    results = []
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y+h, x:x+w]
        eyes = self.eye_cascade.detectMultiScale(roi_gray)
        if len(eyes) >= 2:
            cropped_face = img[y:y+h, x:x+w]
            results.append((cropped_face, (x, y, w, h)))

    if results:
        return results
    else:
        return []

def preprocess_image(self, image):
    """Покращена попередня обробка зображення"""
    # Нормалізація освітлення
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,
8))
    enhanced = clahe.apply(gray)

    # Гауссовий фільтр для зменшення шуму
    enhanced = cv2.GaussianBlur(enhanced, (3, 3), 0)

    # Повертаємо до кольорового формату
    enhanced_color = cv2.cvtColor(enhanced,
cv2.COLOR_GRAY2BGR)
    return enhanced_color

def w2d(self, img, mode='db4', level=3):
    """Покращена версія з більш складним вейвлетом"""
    if len(img.shape) == 3:
        imArray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

```

```

else:
    imArray = img
    imArray = np.float32(imArray) / 255.0
    coeffs = pywt.wavedec2(imArray, mode, level=level)

    # Створюємо кілька варіантів для більш багатих ознак
    coeffs_H = list(coeffs)
    coeffs_H[0] *= 0 # Обнуляємо апроксимацію

    imArray_H = pywt.waverec2(coeffs_H, mode)
    imArray_H *= 255
    imArray_H = np.uint8(np.clip(imArray_H, 0, 255))
    return imArray_H

def extract_lbp_features(self, image, radius=1, n_points=8):
    """Витягування локальних бінарних патернів"""
    if not LBP_AVAILABLE:
        return np.array([])

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) if
len(image.shape) == 3 else image
    lbp = local_binary_pattern(gray, n_points, radius,
method='uniform')
    hist, _ = np.histogram(lbp.ravel(), bins=n_points + 2,
density=True)
    return hist

def extract_statistical_features(self, image):
    """Витягування простих статистичних ознак"""
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) if
len(image.shape) == 3 else image

    # Основні статистики
    mean_val = np.mean(gray)
    std_val = np.std(gray)
    min_val = np.min(gray)
    max_val = np.max(gray)

    # Градієнти (замість GLCM)
    grad_x = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
    grad_y = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)

    grad_mag = np.sqrt(grad_x ** 2 + grad_y ** 2)
    grad_mean = np.mean(grad_mag)
    grad_std = np.std(grad_mag)

    # Лапласіан для виявлення країв
    laplacian = cv2.Laplacian(gray, cv2.CV_64F)
    lap_var = np.var(laplacian) # Міра різкості

    return np.array([mean_val, std_val, min_val, max_val,
grad_mean, grad_std, lap_var])

```

```

def extract_enhanced_features(self, image, target_size=(64,
64)):
    """Витягування розширеного набору ознак"""
    # Масштабування
    scaled_img = cv2.resize(image, target_size)

    # 1. Сирі пікселі (зменшений розмір для швидкості)
    small_img = cv2.resize(image, (32, 32))
    raw_features = small_img.flatten()

    # 2. Вейвлет ознаки (кілька типів)
    wavelet_db4 = self.w2d(image, 'db4', 3)
    wavelet_db4_scaled = cv2.resize(wavelet_db4, (32, 32))

    wavelet_haar = self.w2d(image, 'haar', 2)
    wavelet_haar_scaled = cv2.resize(wavelet_haar, (32, 32))

    # 3. LBP ознаки (якщо доступні)
    lbp_features = self.extract_lbp_features(image)

    # 4. Статистичні ознаки
    stat_features = self.extract_statistical_features(image)

    # 5. Гістограми кольорів
    hist_b = cv2.calcHist([scaled_img], [0], None, [16], [0,
256]).flatten()
    hist_g = cv2.calcHist([scaled_img], [1], None, [16], [0,
256]).flatten()
    hist_r = cv2.calcHist([scaled_img], [2], None, [16], [0,
256]).flatten()

    # 6. HSV гістограми для кращого представлення кольору
    hsv_img = cv2.cvtColor(scaled_img, cv2.COLOR_BGR2HSV)
    hist_h = cv2.calcHist([hsv_img], [0], None, [16], [0,
180]).flatten()
    hist_s = cv2.calcHist([hsv_img], [1], None, [16], [0,
256]).flatten()
    hist_v = cv2.calcHist([hsv_img], [2], None, [16], [0,
256]).flatten()

    # Комбінування всіх ознак
    feature_list = [
        raw_features / 255.0, # Нормалізація
        wavelet_db4_scaled.flatten() / 255.0,
        wavelet_haar_scaled.flatten() / 255.0,
        stat_features / 255.0, # Нормалізація статистичних
ознак
        hist_b / (np.sum(hist_b) + 1e-7), # Нормалізовані
гістограми
        hist_g / (np.sum(hist_g) + 1e-7),
        hist_r / (np.sum(hist_r) + 1e-7),
        hist_h / (np.sum(hist_h) + 1e-7),
        hist_s / (np.sum(hist_s) + 1e-7),
        hist_v / (np.sum(hist_v) + 1e-7)

```

```
]
# Додаємо LBP ознаки, якщо доступні
if len(lbp_features) > 0:
    feature_list.append(lbp_features)
combined_features = np.concatenate(feature_list)
return combined_features
```

Додаток Б

Програмна реалізація модуля візуалізації результатів класифікації

```
import os
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
class ModelEvaluationVisualizer:
    """Клас для збереження візуалізацій результатів
    класифікації"""
    def __init__(self, save_path='results'):
        self.save_path = save_path
        if not os.path.exists(save_path):
            os.makedirs(save_path)
            print(f"Створено теку: {save_path}")

    def save_confusion_matrix(self, y_true, y_pred, class_names,
model_name):
        """
        Збереження матриці плутанини як зображення.
        :param y_true: Справжні мітки
        :param y_pred: Передбачені мітки
        :param class_names: Список назв класів
        :param model_name: Назва моделі
        """
        cm = confusion_matrix(y_true, y_pred)
        plt.figure(figsize=(10, 8))
        sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                    xticklabels=class_names,
yticklabels=class_names)
        plt.title(f'Матриця плутанини - {model_name}',
fontsize=16)
        plt.xlabel('Предбачений клас', fontsize=12)
        plt.ylabel('Справжній клас', fontsize=12)
        plt.tight_layout()
        filename = os.path.join(self.save_path,
f'confusion_matrix_{model_name.lower().replace(" ", "_")}.png')
        plt.savefig(filename, dpi=300, bbox_inches='tight')
        plt.close()
        print(f"[✓] Матрицю плутанини збережено: {filename}")

    def save_accuracy_plot(self, model_accuracies):
        models = list(model_accuracies.keys())
        accuracies = list(model_accuracies.values())
        plt.figure(figsize=(12, 8))
        bars = plt.bar(models, accuracies, color=['skyblue',
'lightcoral', 'lightgreen', 'gold'])
        for bar, acc in zip(bars, accuracies):
            height = bar.get_height()
            plt.text(bar.get_x() + bar.get_width() / 2., height +
0.01,
```

```
        f'{acc:.4f}', ha='center', va='bottom',
fontsize=12, fontweight='bold')
    plt.title('Порівняння точності моделей класифікації',
fontsize=16, fontweight='bold')
    plt.xlabel('Модель', fontsize=12)
    plt.ylabel('Точність', fontsize=12)
    plt.ylim(0, 1.1)
    plt.grid(axis='y', alpha=0.3)
    best_idx = accuracies.index(max(accuracies))
    bars[best_idx].set_color('orange')
    bars[best_idx].set_edgecolor('red')
    bars[best_idx].set_linewidth(2)
    plt.tight_layout()
    filename = os.path.join(self.save_path,
'model_accuracy_comparison.png')
    plt.savefig(filename, dpi=300, bbox_inches='tight')
    plt.close()
```

Додаток В

Програмна реалізація графічного веб-інтерфейсу для завантаження зображень та класифікації

```
import gradio as gr
from PIL import Image, ImageDraw, ImageFont
import numpy as np
import cv2
import joblib
from FaceDetection.feature_face_detection import FeatureFaceDetector

custom_theme = (gr.themes.Base(
    primary_hue="blue",
    secondary_hue="gray",
    font=[gr.themes.GoogleFont("Open Sans"), "sans-serif"]
))

results_dir = "./results3/"

model = joblib.load(f"{results_dir}best_face_classifier.pkl")
scaler = joblib.load(f"{results_dir}scaler.pkl")
class_dict = joblib.load(f"{results_dir}class_dictionary.pkl")

featureFaceDetector = FeatureFaceDetector(effects='rectangle')

def predict_identity(image):
    if isinstance(image, Image.Image):
        image = np.array(image)

        results = featureFaceDetector.get_cropped_images_if_2_eyes(image)

        if not results:
            return image, "Облич з 2 очима не знайдено"

        output_img = image.copy()
        predictions = []

        for cropped_face, (x, y, w, h) in results:
            try:
                features = featureFaceDetector.extract_enhanced_features(cropped_face).reshape(1, -1)
                prediction = model.predict(features)[0]
                class_name = [k for k, v in class_dict.items() if v == prediction][0]
                predictions.append(class_name)
```

```

        cv2.rectangle(output_img, (x, y), (x+w, y+h), (0, 255,
0), 2)
        cv2.putText(output_img, class_name, (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (36,255,12), 2)

    except Exception as e:
        predictions.append("Помилка")

    return output_img, f"Передбачено: {' '.join(predictions)}"

with gr.Blocks(theme=custom_theme, title="Модуль виділення ознак
зображень") as iface:
    gr.Markdown("""
    ### Дипломна робота: Програмний модуль виділення ознак зображень
для покращення точності класифікації в системах комп'ютерного зору
**Автор**: ФЕДЧИШИН Вікторія Сергіївна
    ---
    Завантажте зображення для аналізу. Встановіть параметри нижче.
Натисніть **Запустити аналіз**.
    """)

    with gr.Row():
        with gr.Column(scale=1):
            img_input = gr.Image(type="pil", label="Завантажити
зображення")
            btn = gr.Button("Запустити аналіз")
        with gr.Column(scale=1):
            outputImage = gr.Image(type="pil")
            outputText = gr.Text(label="Результат обробки")
        btn.click(fn=predict_identity, inputs=[img_input],
outputs=[outputImage, outputText])

if __name__ == "__main__":
    iface.launch()

```

Додаток Г
Копія публікації

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ПТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Степаник Олександр, Ліп'яніна-Гончаренко Христина	86
МЕТОД ІДЕНТИФІКАЦІЇ ОЗНАК ТВАРИН НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ	86
Ткачишин Віктор, Дорош Віталій	89
АДАПТИВНА СИСТЕМА ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ У СЕРЕДОВИЩІ ВЕЛИКИХ ДАНИХ	89
Трубач Михайло, Дорош Віталій	93
ОПТИМІЗАЦІЯ СТРУКТУРИ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ALEXNET ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ	93
Федчишин Вікторія, Биковий Павло	97
ПРОГРАМНИЙ МОДУЛЬ ВИДІЛЕННЯ ОЗНАК ЗОБРАЖЕНЬ ДЛЯ ПОКРАЩЕННЯ ТОЧНОСТІ КЛАСИФІКАЦІЇ В СИСТЕМАХ КОМП'ЮТЕРНОГО ЗОРУ	97
Черемшинський Андрій, Домбровський Михайло	100
МОДУЛЬ ПРОГНОЗУВАННЯ ВПЛИВУ КРИЗОВИХ СИТУАЦІЙ НА ЕНЕРГОСПОЖИВАННЯ НА ОСНОВІ ЧАСОВИХ РЯДІВ	100
Штинда Віктор, Чіп Святослав, Козак Аліна	103
ОГЛЯД СУЧАСНИХ МОДЕЛЕЙ У ЗАДАЧАХ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ І ТЕКСТІВ	103
СЕКЦІЯ 2. ІНТЕЛЕКТУАЛЬНІ ІТ В ОСВІТІ, КУЛЬТУРІ ТА ГУМАНІТАРНИХ НАУКАХ	106
Боднар Анатолій, Лендюк Тарас	106
МОДУЛЬ КЛАСИФІКАЦІЇ СПАМУ В SMS-ПОВІДОМЛЕННЯХ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ NLTK	106
Божагора Микола, Дорош Віталій	108
МОДУЛЬ РОЗПІЗНАВАННЯ СТАТІ НА ОСНОВІ ГЛИБОКОЇ НЕЙРОМЕРЕЖЕВОЇ АРХІТЕКТУРИ INSERTIONV3	108
Бугера Назарій	111
ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗУМНОГО ТУРИЗМУ НА ОСНОВІ ТЕХНОЛОГІЙ БЛОКЧЕЙН ТА ВЕЛИКИХ ДАНИХ	111
Буднік Сергій, Ніпіалді Ольга	114
ОЦІНЮВАННЯ ВАРТОСТІ ІНТЕРНЕТ-ПЛАТФОРМ НА ОСНОВІ ТЕХНОЛОГІЇ ВЕЛИКИХ ДАНИХ	114
Вархів Богдан, Осолінський Олександр	116
ІНФОРМАЦІЙНА СИСТЕМА ІНТЕГРОВАНИХ МОДУЛІВ ЗНАНЬ ІОТ	116

Федчишин Вікторія
студентка групи КН-42
v.fedchyshyn@st.wunu.edu.ua

Биковий Павло
к.т.н., доцент
pb@wunu.edu.ua

Західноукраїнський національний університет
Тернопіль, Україна

ПРОГРАМНИЙ МОДУЛЬ ВИДІЛЕННЯ ОЗНАК ЗОБРАЖЕНЬ ДЛЯ ПОКРАЩЕННЯ ТОЧНОСТІ КЛАСИФІКАЦІЇ В СИСТЕМАХ КОМП'ЮТЕРНОГО ЗОРУ

У сучасних промислових системах візуального контролю класифікація об'єктів зображення значною мірою залежить від якісного виділення ознак. Саме цей етап формує інформативне уявлення про зображення, зменшує його розмірність і підвищує ефективність роботи моделей машинного навчання [1].

У межах даної роботи створено програмний модуль, який виконує послідовну обробку зображень: виявлення обличчя, перевірку наявності двох очей, обрізання області обличчя, трансформацію зображення за допомогою вейвлет-перетворення (Wavelet Transform) та класифікацію з використанням методу опорних векторів (SVM). Структуру модуля подано на рисунку 1.

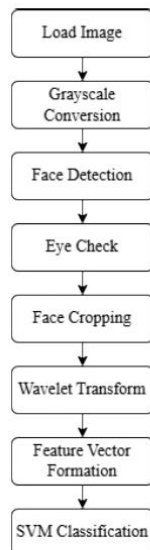


Рисунок 1 – Структура програмного модуля обробки зображень

На етапі попередньої обробки зображення перетворюються в градації сірого та перевіряються на наявність двох очей за допомогою класифікаторів Haar. У випадку позитивного результату виділяється область обличчя, яка далі піддається вейвлет-перетворенню (Wavelet Transform, тип db1, рівень 5) для виявлення текстурних ознак. Обидві частини ознак (масштабоване обрізане зображення без трансформацій та його вейвлет-перетворення) об'єднуються у єдиний вектор ознак [2].

Модель класифікації побудовано за допомогою SVM з радіальним базисним ядром (RBF). Для тренування використано власний датасет зображень публічних осіб, попередньо структурований та оброблений. Було сформовано близько 400 зображень для навчання та тестування, поділених на класи відповідно до імені об'єкта на фото.

Попередні дослідження демонстрували, що подібні підходи забезпечують точність понад 85% [3]. У таблиці 1 наведено орієнтовні результати роботи моделі на основі комбінованого підходу.

Таблиця 1 – Результати класифікації моделі Haar + Wavelet + SVM

Метод	Точність (%)	Повнота (%)	F1-міра (%)
Haar + Wavelet + SVM	85.2	83.5	84.3

З отриманих результатів видно, що навіть без використання глибоких нейронних мереж якісне попереднє виділення ознак дозволяє досягти високої точності класифікації. Перевагою запропонованого підходу є нижчі вимоги до обчислювальних ресурсів, що особливо актуально для вбудованих систем та промислових рішень в умовах обмеженої апаратної бази.

Після цього застосовуються методи виділення ознак, зокрема трансформація за допомогою вейвлетів, що дозволяє отримати детальні текстурні характеристики. Одним із прикладів класичних методів виділення ознак є використання градієнтних фільтрів, таких як оператор Собеля, який виявляє горизонтальні та вертикальні краї об'єктів на зображенні. Цей підхід є базовим прикладом попереднього аналізу зображення перед побудовою моделей класифікації [2].

У сучасних дослідженнях активно вивчається ефективність використання вейвлет-перетворень для покращення виділення ознак у задачах класифікації зображень. Зокрема, поєднання вейвлет-декомпозиції з класичними методами машинного навчання, такими як SVM, демонструє високі результати при обмежених обчислювальних ресурсах. Наприклад, у роботі [3] запропоновано гібридну модель, що поєднує вейвлет-декомпозицію з EfficientNet для класифікації зображень шкіри, досягаючи точності до 94,7%. Це підтверджує ефективність

вейвлет-перетворень у виділенні текстурних ознак, що є критичним для точності класифікації.

Крім того, дослідження [4] показує, що використання вейвлет-перетворень у поєднанні з CNN дозволяє зменшити обчислювальну складність без втрати точності, що особливо важливо для систем з обмеженими ресурсами. Ці підходи демонструють, що навіть без використання глибоких нейронних мереж, якісне попереднє виділення ознак може забезпечити високу точність класифікації.

Розроблений модуль легко інтегрується у системи контролю якості продукції, наприклад, у машинобудуванні, харчовій або електронній промисловості. Його реалізація відповідає принципам Industry 4.0, що передбачає автономні інтелектуальні рішення, здатні до адаптації на основі вхідних даних [5].

Список використаних джерел

1. Feature Extraction and Image Classification using OpenCV. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/feature-extraction-and-image-classification-using-opencv/> (дата звернення: 26.05.2025).
2. The Role of Feature Extraction in Machine Learning. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/the-role-of-feature-extraction-in-machine-learning/> (дата звернення: 26.05.2025).
3. Aboulmira A., Hrimch H., Lachgar M., Hanine M., Garcia C. O., Mezquita G. M., Ashraf I. Hybrid Model with Wavelet Decomposition and EfficientNet for Accurate Skin Cancer Classification. *Journal of Cancer*. 2025. Vol. 16, No. 2. P. 506–520. DOI: [10.7150/jca.101574](https://doi.org/10.7150/jca.101574).
4. Dede M., Yildirim M. Wavelet-Based Feature Extraction for Efficient High-Resolution Image Classification. *Engineering Reports*. 2025. Article e70027. DOI: [10.1002/eng2.70027](https://doi.org/10.1002/eng2.70027).
5. Liu C., Zhou Y., Qian Y. Industrial Visual Inspection based on Lightweight CNNs: A Review. *IEEE Transactions on Industrial Informatics*. 2022. Vol. 18, No. 5. P. 3172–3181. DOI: [10.1109/TII.2021.3110892](https://doi.org/10.1109/TII.2021.3110892).