

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
Циклова комісія (кафедра) комп'ютерної інженерії та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ

Виконав: студент групи 2П-21

Спеціальності

121 Інженерія програмного забезпечення

Данил ВІЛЯМОВСЬКИЙ

Керівник:

Маргарита МЕДОЛИЗ

Черкаси 2025

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена розробці програмного забезпечення для розпізнавання облич. У роботі проведено огляд поточного стану предметної області, включаючи загальні підходи до розпізнавання облич, аналіз алгоритмів та моделей, що застосовуються в сучасних системах, а також існуючих програмних рішень. На основі проведеного аналізу сформульовано постановку задачі. В рамках проєктування та реалізації програмного продукту здійснено аналіз вимог до розроблюваного програмного забезпечення, що стало основою для створення ефективного рішення у сфері розпізнавання облич.

ABSTRACT

This qualification paper is dedicated to the development of software for face recognition. The work provides an overview of the current state of the art in the subject area, including general approaches to face recognition, an analysis of algorithms and models applied in modern systems, and existing software solutions. Based on the conducted analysis, the problem statement is formulated. Within the scope of software product design and implementation, an analysis of the requirements for the developed software was performed, which served as the basis for creating an effective solution in the field of face recognition.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1 ОГЛЯД ПОТОЧНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Загальні підходи до розпізнавання облич.....	7
1.2 Огляд алгоритмів та моделей, що застосовуються в сучасних системах розпізнавання.....	13
1.3 Аналіз програмних рішень для розпізнавання облич.....	18
1.4 Формулювання постановки задачі	23
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОЄКТУ	25
2.1 Аналіз вимог до програмного забезпечення, що розробляється.....	25
2.2 Моделювання предметної області.....	29
2.3 Програмна реалізація проєкту	33
РОЗДІЛ 3 ТЕСТУВАННЯ ТА СУПРОВОДЖЕННЯ	39
3.1 Перелік і обґрунтування обраних методів тестування.....	39
3.2 Тестовий план проєкту	40
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47

ВСТУП

У сучасному світі технології розпізнавання облич активно впроваджуються у системи безпеки, контролю доступу, комерційні сервіси, банківські структури та мобільні пристрої. Згідно з аналітичними дослідженнями, глобальний ринок програмного забезпечення для розпізнавання облич зростає щорічно в середньому на 17,2% і досягне \$12,67 млрд до 2028 року [1]. Це свідчить про високий попит на точні, надійні та масштабовані системи і, відповідно, про актуальність розробки ефективного інструменту для ідентифікації особи за зображенням.

Проблема полягає у забезпеченні високої точності розпізнавання облич в умовах змін освітлення, повороту голови, наявності перешкод (наприклад, окулярів або масок), а також у зменшенні обчислювальних витрат при реалізації таких систем на вбудованих або мобільних пристроях. Більшість наявних рішень базуються на глибоких нейронних мережах, зокрема згорткових нейронних мережах (CNN), однак постає задача оптимізації таких моделей для використання в реальному часі без втрати точності.

Методи розпізнавання облич активно досліджуються вченими протягом останніх десятиліть. Основоположні підходи, зокрема метод Eigenfaces [2], Fisherfaces [3] та алгоритм Viola-Jones [4], стали базисом для подальшої еволюції систем комп'ютерного зору. Надалі з розвитком глибокого навчання були запропоновані високоточні архітектури, наприклад, DeepFace [5], FaceNet [6], VGGFace [7] та ArcFace [8]. Незважаючи на значний прогрес, існує потреба у розробці адаптивного, ресурсозберігаючого та точного ПЗ, яке може бути інтегроване у прикладні рішення.

Об'єктом дослідження є процес автоматичного розпізнавання облич на зображеннях і відео.

Предметом дослідження є алгоритмічні методи та інструменти комп'ютерного зору, які забезпечують точну ідентифікацію облич у різних

умовах з використанням програмних бібліотек OpenCV, Dlib, TensorFlow та PyTorch.

Мета роботи полягає в розробці програмного забезпечення для розпізнавання облич з використанням сучасних алгоритмів комп'ютерного зору, яке забезпечує точну ідентифікацію особи за зображенням у режимі, близькому до реального часу.

Завдання дослідження:

- проаналізувати існуючі методи та бібліотеки для розпізнавання облич;
- здійснити порівняння ефективності різних моделей;
- розробити архітектуру програмного забезпечення з урахуванням обраного алгоритму;
- реалізувати програмне забезпечення для розпізнавання облич;
- провести експериментальну перевірку точності та продуктивності реалізованої системи.

Практичне значення роботи полягає в розробці програмної системи, здатної не лише до розпізнавання облич, а й до детального аналізу зображень для визначення метричних характеристик обличчя, таких як симетрія, пропорції, форма, площа окремих ділянок тощо.

Отримані результати можуть застосовуватись у медичній практиці — зокрема в дерматології, пластичній хірургії, телемедицині, ортодонтії та психофізіології — для діагностики аномалій розвитку, виявлення асиметрій, моніторингу післяопераційного стану або аналізу вікових змін.

Система має модульну архітектуру, що дозволяє інтегрувати її в існуючі медичні інформаційні комплекси, зокрема з можливістю поєднання з електронною медичною картою пацієнта. Використання глибоких нейронних мереж забезпечує високу точність оцінки параметрів облич та стійкість до варіацій зображень, що робить систему придатною як для діагностичних, так і для дослідницьких цілей.

РОЗДІЛ 1

ОГЛЯД ПОТОЧНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні підходи до розпізнавання облич

Розпізнавання облич (face recognition) є одним із провідних напрямів у галузі комп'ютерного зору та біометричної ідентифікації. Це завдання передбачає автоматизоване визначення особи за її візуальним образом та широко застосовується в системах безпеки, фінансових технологіях, побутових пристроях та урядових інформаційних системах. Як прикладна проблема, розпізнавання облич охоплює кілька послідовних етапів: попередню обробку зображення, виявлення облич, побудову дескрипторів (ознак) і класифікацію об'єкта. Основні поняття в межах цієї задачі включають виявлення облич (face detection), що полягає в локалізації області з обличчям на зображенні або відеопотоці; ідентифікацію (identification), яка передбачає визначення особи серед множини зареєстрованих у базі даних; а також верифікацію (verification), що полягає у підтвердженні належності обличчя до конкретного іменного запису [1, с. 219].

У рамках більш загальної структури комп'ютерного зору, розпізнавання облич тісно пов'язане із суміжними задачами, такими як класифікація об'єктів, відстеження руху (трекінг), сегментація зображень та семантична інтерпретація сцени [2, с. 15]. Загальна послідовність операцій, яка забезпечує функціонування типової системи розпізнавання облич, містить кілька ключових етапів. Перший етап – це виявлення обличчя, що реалізується за допомогою алгоритмів, таких як Viola-Jones, Haar cascade або сучасних глибоких моделей на основі багат шарових згорткових мереж, зокрема МТСNN [3]. На наступному етапі відбувається нормалізація зображення: обличчя вирівнюється за рівнем очей, усуваються відмінності в освітленні, зображення масштабується до стандартного розміру, що забезпечує стабільність подальшої обробки. Після нормалізації виконується витягування ознак перетворення області обличчя у вектор

характеристик, який є унікальним і стійким до змін пози, освітлення та інших зовнішніх факторів. У сучасних системах для цього використовують як класичні підходи (локальні бінарні шаблони – LBP, гістограми направлених градієнтів – HOG), так і нейронні мережі з глибокою архітектурою (наприклад, VGG-Face, FaceNet, ArcFace), які навчаються безпосередньо на великих наборах облич [4].

Остаточний етап – це класифікація, суть якої полягає в порівнянні вектора ознак із базою відомих шаблонів для визначення ідентичності особи. На цьому етапі застосовуються різноманітні класифікатори: від традиційних алгоритмів, на зразок методу опорних векторів (SVM), методів найближчих сусідів (k-NN) до глибоких моделей із Softmax-виходом або metric-learning з використанням косинусної відстані [5]. Узагальнена блок-схема процесу розпізнавання облич із зазначенням умовних переходів і послідовності обробки представлена на рисунку 1.1. Вона демонструє основні етапи: від завантаження зображення і попередньої обробки до класифікації, а також обробку помилкових сценаріїв, наприклад, у випадку, коли обличчя не знайдено на вхідному зображенні.

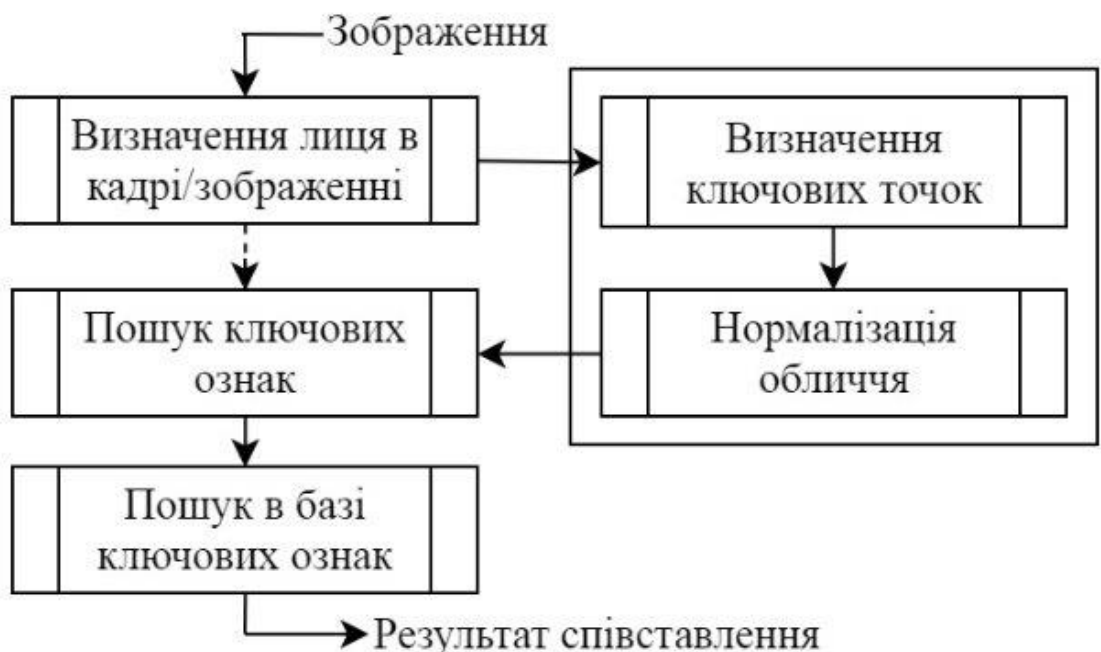


Рис.1.1 - Блок-схема процесу розпізнавання облич

Методи розпізнавання облич за час свого розвитку пройшли значну еволюцію: від простих геометричних моделей до складних глибоких нейронних

мереж, здатних до самостійного навчання на великих обсягах даних. Загальноприйнято класифікувати ці методи за принципами, що лежать в основі їхньої роботи: геометричні, шаблонні, статистичні, машинного навчання та нейронні.

Геометричні методи розпізнавання ґрунтуються на припущенні, що просторові відношення між ключовими точками на обличчі (наприклад, між очима, носом, ротом) є унікальними для кожної особи. У цих методах обличчя моделюється як сукупність маркерів, і розпізнавання відбувається шляхом порівняння набору відстаней або кутів між ними з шаблонами з бази даних [6]. Такий підхід є простим у реалізації, але чутливим до поворотів голови, виразу обличчя та якості зображення.

Методи на основі шаблонів (template-based) передбачають зіставлення зображення обличчя з еталонними шаблонами за допомогою метрик подібності, таких як евклідова чи кореляційна відстань. Вони забезпечують високу точність при умовах однакового масштабу та орієнтації, але значно знижують продуктивність при наявності шуму чи змін умов зйомки [7].

Алгоритми на основі статистичних моделей, такі як метод головних компонент (PCA, або Eigenfaces) і лінійний дискримінантний аналіз (LDA, або Fisherfaces), дозволяють зменшити розмірність зображення і виділити найбільш інформативні ознаки. Метод PCA виділяє компоненти з найвищою дисперсією, тоді як LDA спрямований на максимальну відмінність між класами [8]. Ці методи ефективні для невеликих баз даних і добре працюють за умови нормалізованого вводу.

Сучасні методи машинного навчання (SVM, Random Forest, k-NN) використовуються для класифікації векторів ознак, отриманих із зображень облич. Вони дозволяють побудувати моделі з високою узагальнюючою здатністю, проте залежать від якості ознак і кількості тренувальних зразків [9].

Найбільш передовим напрямом сьогодні є застосування глибоких нейронних мереж, зокрема згорткових нейронних мереж (CNN) та сіамських мереж (Siamese networks). Мережі CNN дають змогу автоматично витягувати

релевантні ознаки з вхідного зображення і навчатися класифікувати обличчя з великою точністю [10]. Сіамські мережі, навпаки, не класифікують безпосередньо, а обчислюють відстань між двома зображеннями, що дозволяє ефективно реалізувати верифікацію навіть при невеликій кількості прикладів на особу [11]. Загальну класифікацію методів розпізнавання обличчя подано в таблиці 1.1.

Таблиця 1.1 – Класифікація методів розпізнавання обличчя

Клас методів	Основний принцип роботи	Приклади/алгоритми	Переваги	Недоліки
Геометричні	Відстані між ключовими точками	Point-distance model	Простота, швидкість	Залежність від пози та освітлення
На основі шаблонів	Зіставлення з еталонними зображеннями	Template Matching	Пряма реалізація	Чутливість до масштабу і шумів
Статистичні моделі	Зменшення розмірності, виділення інформативних ознак	PCA, LDA	Ефективність при малих базах	Складність у масштабуванні
Машинне навчання	Навчання класифікаторів на векторах ознак	SVM, Random Forest	Гнучкість, адаптивність	Потреба в якісних ознаках
Глибокі нейронні мережі	Автоматичне витягнення ознак і класифікація	CNN, FaceNet, ArcFace	Висока точність, стійкість до шумів	Висока обчислювальна складність
Сіамські нейронні мережі	Обчислення подібності між зображеннями	Siamese, Triplet networks	Працює з мінімумом прикладів	Ускладнене навчання, потреба в парних даних

Згідно з оглядом сучасної літератури [12], глибоке навчання витісняє інші методи у високоточних застосуваннях, особливо у великих відкритих наборах даних (наприклад, LFW, VGGFace2). Проте у вбудованих або ресурсообмежених

середовищах досі можуть ефективно використовуватися статистичні або гібридні підходи.

До появи глибоких нейронних мереж основними підходами до розпізнавання облич були методи, побудовані на статистичному аналізі зображень. Одним із перших ефективних методів став підхід Eigenfaces, заснований на методі головних компонент (PCA). Цей метод передбачає перетворення зображення обличчя у вектор і його проєкцію на простір головних компонент із максимальною дисперсією. Перевага PCA полягає у зменшенні розмірності та виявленні загальних закономірностей, однак він чутливий до зміни освітлення та пози [13].

Метод Fisherfaces, побудований на лінійному дискримінантному аналізі (LDA), орієнтований на максимізацію міжкласових відмінностей при мінімізації внутрікласової дисперсії. У порівнянні з Eigenfaces, Fisherfaces краще справляється з задачами класифікації в умовах змін освітлення, але вимагає великої кількості навчальних прикладів для кожного класу [14].

Іншим важливим підходом є локальні бінарні шаблони (LBP) – метод аналізу текстури, в якому зображення розглядається як сукупність локальних вікон, у межах яких формується бінарний шаблон на основі порівняння інтенсивності пікселів. LBP є обчислювально простим, нечутливим до змін яскравості та придатним до роботи в реальному часі, однак поступається у точності глибоким методам [15].

Класичні підходи мають спільну перевагу – низькі обчислювальні вимоги, що дозволяє їх використання у вбудованих системах. Проте їх ефективність суттєво знижується в умовах варіативності даних (повороти, **емоції**, освітлення), що обмежує можливості масштабування таких систем у складних реальних сценаріях.

З появою потужних обчислювальних засобів і великих відкритих наборів даних почала активно розвиватися технологія розпізнавання облич на основі згорткових нейронних мереж (CNN). Ці моделі здатні автоматично витягувати

складні, ієрархічні ознаки без попереднього ручного проєктування дескрипторів, що значно підвищує ефективність системи в умовах реальних даних.

Першим проривом у галузі стала модель DeepFace від Facebook, яка використовувала дев'ятишарову згорткову архітектуру та тривимірне вирівнювання зображень облич, що дало змогу досягти точності, близької до людської [16]. Наступним кроком стала розробка моделі FaceNet (Google), де вперше застосовано метричне навчання із триплетною втратою, що дозволяє створювати векторне представлення зображення з властивістю відстані як критерію подібності [17]. Інші архітектури, зокрема VGGFace, а також ArcFace, запропонували покращену геометричну інтерпретацію простору ознак, зокрема введення кутових обмежень для підвищення міжкласової відстані і внутрікласової компактності [18].

Залежно від поставленої задачі, сучасні моделі використовують два базові підходи до навчання. У класифікаційному навчанні мережа навчається визначати ймовірність приналежності зображення до одного з фіксованих класів, що добре працює при наявності повної бази користувачів. Натомість метричне навчання не потребує фіксованих класів, а навчає модель оцінювати подібність між парами або трійками прикладів, що забезпечує масштабованість і універсальність у задачах відкритої ідентифікації. Основні характеристики цих підходів узагальнено в таблиці 1.2.

Таблиця 1.2 – Порівняння класифікаційного та метричного навчання у глибоких моделях

Параметр	Класифікаційне навчання	Метричне навчання
Принцип	Навчання на фіксованих класах	Оцінка подібності між прикладами
Приклади моделей	DeepFace, VGGFace	FaceNet, ArcFace, Siamese
Вихід	Вектор ймовірностей	Вектор ознак у метричному просторі
Потреба у фіксованій базі	Так	Ні

Продовження таблиці 1.1

Гнучкість при розширенні	Обмежена	Висока
Застосування	Системи з фіксованими користувачами	Масштабовані системи, відкриті набори
Складність навчання	Низька–середня	Висока (потребує генерації пар або трійок)

Однак, незважаючи на точність, глибокі моделі мають проблеми перенавчання на конкретних наборах даних, що проявляється у зниженні ефективності при перенесенні на нові домени. Також спостерігається складність узагальнення, особливо за умови обмеженості або нерепрезентативності тренувальної вибірки. Це знижує адаптивність систем у нестандартних умовах без додаткового донавчання, що важливо враховувати при проектуванні масштабованих і надійних рішень у сфері розпізнавання облич.

1.2 Огляд алгоритмів та моделей, що застосовуються в сучасних системах розпізнавання

Одним із ключових етапів побудови системи розпізнавання облич є виявлення області обличчя на зображенні та його нормалізація. Від якості цього етапу залежить стабільність витягування ознак і точність подальшої класифікації. Залежно від обчислювальних ресурсів і умов роботи (реальний час, мобільні пристрої, хмарні обчислення), для виявлення облич можуть застосовуватись як класичні евристичні методи, так і глибокі нейромережеві моделі.

Першим широко застосовуваним підходом став алгоритм Віоли-Джонса (Viola-Jones), що використовує каскад бінарних слабких класифікаторів на основі Хагг-подібних ознак та методу Adaboost для виявлення облич у реальному часі [19]. Кожна Хагг-ознака визначається як різниця сум пікселів у прямокутних зонах зображення. Формально, така ознака може бути записана як:

$$f(x) = \sum_{i \in R_1} x_i - \sum_{j \in R_2} x_j, \quad (1.1)$$

де R_1 та R_2 – області з протилежними коефіцієнтами, x_i – значення яскравості пікселя.

Іншим підходом є використання гістограм спрямованих градієнтів (HOG) спільно з SVM-класифікатором. Метод полягає у побудові гістограм орієнтацій градієнтів у локальних вікнах зображення. Нехай G_x і G_y – градієнти по осях X та Y відповідно. Напрямок градієнта θ визначається як:

$$f\theta = \arctg\left(\frac{G_y}{G_x}\right), \quad \|G\| = \sqrt{G_x^2 + G_y^2} \quad (1.2)$$

Узагальнена векторна ознака, побудована з гістограм, подається на вхід SVM, який класифікує область як обличчя/необличчя. Цей підхід реалізований у бібліотеці Dlib і добре працює при сталому освітленні та фронтальних зображеннях [20].

Для забезпечення високої точності у реальних умовах (зміна масштабу, нахил голови, часткові перекриття) застосовуються багаторівневі згорткові нейронні мережі. Зокрема, модель MTCNN (Multi-task Cascaded Convolutional Networks) складається з трьох мереж: P-Net, R-Net, і O-Net. Перша здійснює первинне виявлення кандидатів, друга – фільтрацію, третя – точне уточнення координат і виявлення ключових точок (очі, ніс, кути рота) [21].

Іншою сучасною моделлю є RetinaFace, яка використовує backbone-мережу (ResNet, MobileNet) у поєднанні з контекстними модулями для виявлення облич із точністю до 5 ключових точок. У RetinaFace задача формалізується як оптимізація комбінованої втрати:

$$L = L_{cls} + \lambda_1 L_{box} + \lambda_2 L_{landmark}, \quad (1.3)$$

де L_{cls} – функція втрат класифікації (обличчя/необличчя), L_{box} – регресія рамки, $L_{landmark}$ – регресія координат ключових точок, а λ_1, λ_2 – вагові коефіцієнти [22].

Для підвищення точності вхідні зображення проходять нормалізацію – вирівнювання обличчя за орієнтацією, розміром і освітленням. Найпоширеніший підхід – це геометричне вирівнювання за ключовими точками, наприклад, по координатах очей та рота. При цьому виконується афінне перетворення T , що приводить точки $x \in R^2$ до фіксованих координат:

$$x' = T(x) = Ax + b, \quad (1.4)$$

де $A \in R^{2 \times 2}$ – матриця обертання та масштабування, b – вектор зсуву.

У випадках з сильними поворотами застосовуються 3D-моделі облич (3D Morphable Models), які дозволяють побудувати сітку знімка у тривимірному просторі та спроектувати її у фронтальну площину. Це значно підвищує точність подальшого витягування ознак [23].

У системах автоматизованого розпізнавання облич критичним етапом є витягування ознак, що забезпечує перетворення вхідного зображення в компактне та інформативне векторне представлення. Цей процес визначає точність подальшої класифікації, оскільки саме вектор ознак фіксує унікальні характеристики обличчя, незалежні від умов освітлення, масштабу або ракурсу.

Початково витягування ознак базувалося на класичних підходах. Зокрема, ефективним виявився метод гістограм направлених градієнтів (Histogram of Oriented Gradients, HOG), який дозволяє кодувати контурну структуру зображення шляхом обчислення локальних напрямків градієнтів і побудови гістограм. Цей підхід забезпечує збереження інформації про форму та орієнтацію основних рис обличчя. На рисунку 1.2 представлено приклад побудови HOG-дескриптора для сірого зображення – праве зображення візуалізує орієнтаційні градієнти, що відповідають основним напрямкам границь і контурів об'єкта.

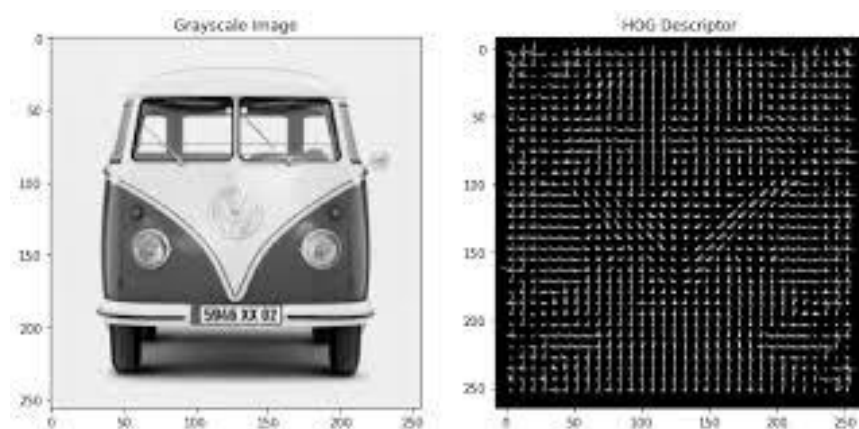


Рисунок 1.2 – HOG-дескриптор для сірого зображення

Інший поширений класичний підхід – локальні бінарні шаблони (Local Binary Patterns, LBP), що аналізують відносну яскравість пікселів у локальному вікні. Результатом є бінарна текстурна маска, з якої формуються гістограми. Ці дескриптори є обчислювально легкими та забезпечують базову стійкість до змін освітлення, проте погано справляються із масштабними варіаціями. Результати такого методу, включаючи оброблені зображення та відповідні гістограми ознак, продемонстровано на рисунку 1.3. Видно, як локальні шаблони дозволяють диференціювати зображення за текстурними властивостями, що є корисним при обмежених ресурсах або у вбудованих системах.

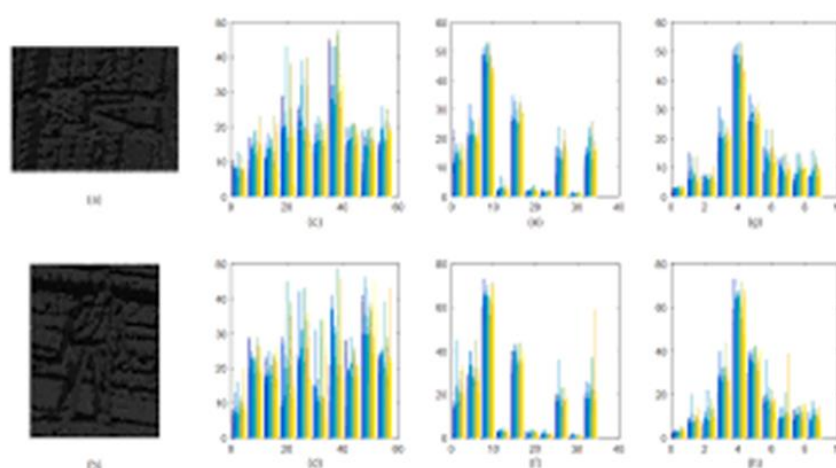


Рисунок 1.3 – Оброблені зображення та відповідні гістограми (LBP)

Сучасні системи розпізнавання орієнтовані на високу точність та використовують згорткові нейронні мережі (CNN), які забезпечують автоматичне витягнення ознак без потреби ручного проектування дескрипторів.

Моделі типу FaceNet, ArcFace чи CosFace формують векторні вбудовування (embeddings), які проєктують зображення в метричний простір фіксованої розмірності. У цьому просторі зображення тієї самої особи розміщуються близько, а різних – на віддалених відстанях. Такі вектори є стійкими до завад, змін пози, часткових перекриттів та варіацій освітлення, а також демонструють високі результати точності на стандартних наборах, таких як LFW, VGGFace2 чи CASIA-WebFace.

Після побудови вектора ознак наступним кроком є класифікація або порівняння обличчя з відомими зразками для визначення особи. Система може бути реалізована як закрита, з фіксованою кількістю класів, або відкрита, де модель має забезпечувати порівняння нових, раніше не бачених осіб. У цьому контексті ключовими є методи класифікації та метричні підходи.

Одним із базових методів класифікації є алгоритм k-найближчих сусідів (k-NN), який ідентифікує об'єкт за належністю до класу, що найчастіше зустрічається серед найближчих векторів у ознаковому просторі. На рисунку 1.4 представлено принцип роботи цього методу: вектор із невідомою міткою позначено зеленим кольором, а найближчі сусіди – червоними трикутниками та синіми квадратами – вказують на приналежність до певного класу на основі більшості.

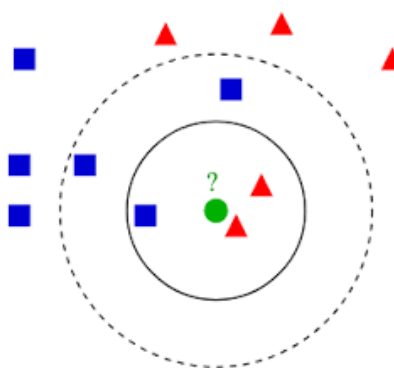


Рисунок 1.4 – Принцип роботи k-NN методу

Метод є інтуїтивно простим і ефективним у випадках з добре відокремленими кластерами ознак, проте його продуктивність знижується при збільшенні розмірності ознакового простору.

Для задач класифікації також широко застосовується метод опорних векторів (SVM), що будує гіперплощину, яка максимізує відстань між класами у векторному просторі. SVM добре працює на малих вибірках і може бути використаний разом із класичними дескрипторами або векторними представленнями CNN. Ще одним ефективним підходом є обчислення косинусної подібності – метрики, яка оцінює кутову близькість між двома векторами. Такий підхід часто використовується в глибоких моделях, орієнтованих на верифікацію, де важливо визначити, наскільки подібні два зображення.

Класифікаційні підходи, зокрема з використанням Softmax-функції, ефективні у закритих системах, де кількість класів (осіб) фіксована. Однак при додаванні нових користувачів потрібне повне перенавчання. У свою чергу, підходи на основі метричного навчання (metric learning), як у моделях FaceNet або ArcFace, дозволяють здійснювати порівняння ознак без прямої класифікації, що робить їх ідеальними для відкритих систем. Вони формують простір, де відстань між векторами відповідає ступеню подібності облич.

Отже, вибір методу класифікації визначається архітектурою системи, вимогами до масштабування, кількістю користувачів та обмеженнями щодо ресурсів. На практиці найчастіше застосовуються гібридні підходи, де попередній відбір кандидатів здійснюється швидкими евристичними методами, а остаточне рішення – за допомогою метричних моделей або CNN-класифікаторів.

1.3 Аналіз програмних рішень для розпізнавання облич

На сьогодні в системах автоматизованого розпізнавання облич активно використовуються відкриті бібліотеки та програмні фреймворки, що реалізують як класичні алгоритми, так і глибокі нейронні моделі. Вибір конкретного інструменту залежить від поставлених вимог до точності, швидкодії, підтримки апаратного прискорення, обсягу даних та сценаріїв розгортання. У цьому

підпункті розглянуто чотири поширені технологічні рішення: OpenCV, Dlib, FaceNet та InsightFace.

Одним із наймасовіших інструментів є бібліотека OpenCV, що підтримує широкий спектр функцій комп'ютерного зору, включно з розпізнаванням облич. В OpenCV реалізовано базові алгоритми виявлення на основі каскадів Хаара, а також сучасні DNN-моделі, які підтримують використання попередньо навчених нейронних мереж через інтерфейси Caffe, TensorFlow та ONNX. Крім того, бібліотека містить реалізацію LBPН-дескрипторів для класифікації облич за текстурою. На рисунку 1.5 представлено головну сторінку OpenCV, де видно масштабність підтримки проєкту, включаючи навчальні ресурси, курси та інструменти для реального використання [27].

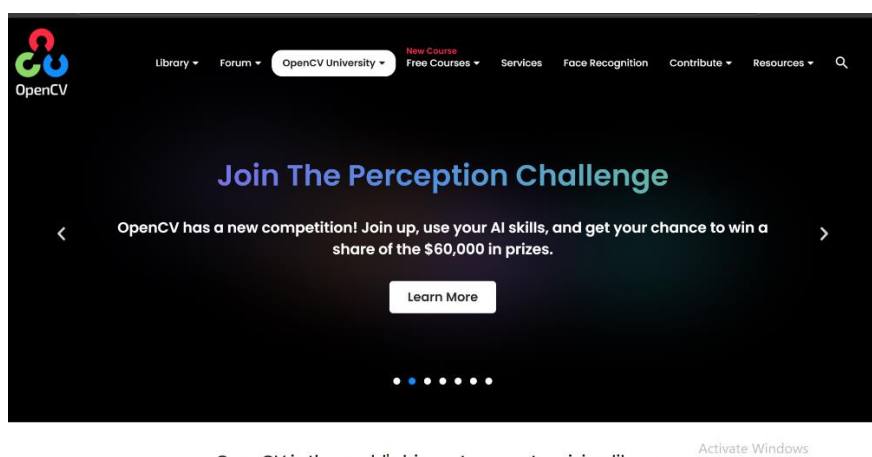


Рисунок 1.5 – Головна сторінка інструменту OpenCV

Іншим ключовим рішенням є Dlib – бібліотека, реалізована мовою C++, з Python-інтерфейсом, яка поєднує методи машинного навчання та комп'ютерного зору. Основною перевагою Dlib є вбудована модель виявлення облич на основі HOG + SVM, а також підтримка глибокої архітектури для вбудовування ознак. Dlib також надає функціонал для вирівнювання облич шляхом знаходження 68 або 5 ключових точок. Бібліотека добре документована, відтестована на різних платформах і може бути використана як у наукових, так і в промислових проєктах. На рисунку 1.6 наведено інтерфейс Dlib C++ Library, який демонструє структурованість функціональних модулів [28].

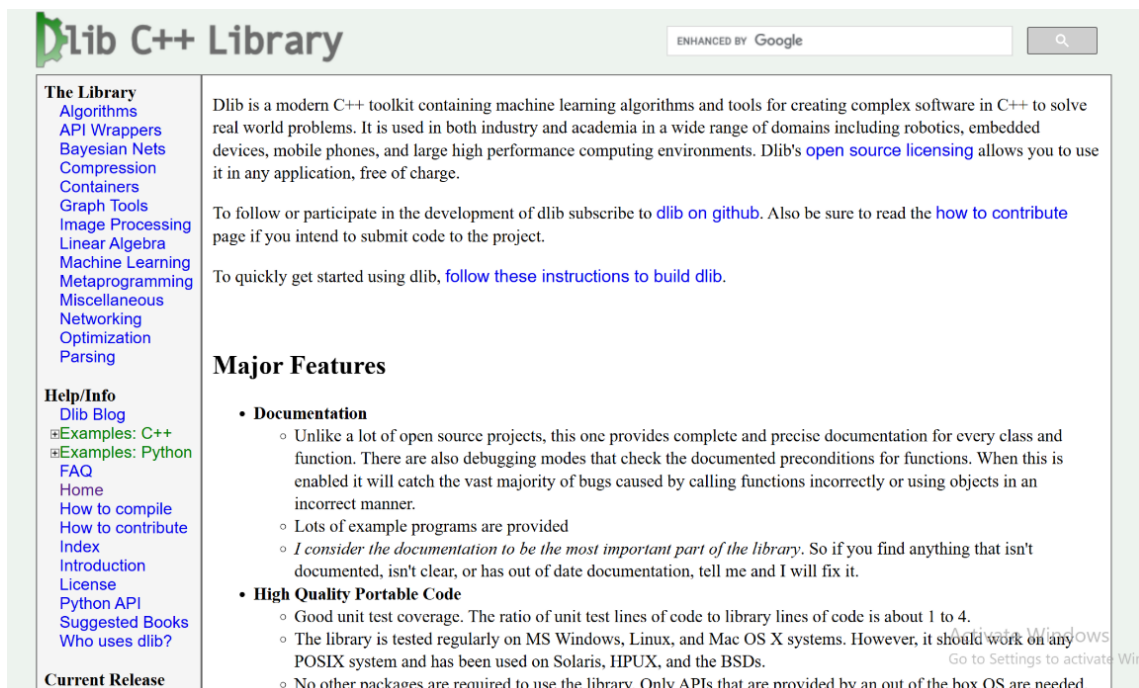


Рисунок 1.6 – Інтерфейс Dlib C++ Library

На рівні глибокого навчання важливу роль відіграє модель FaceNet, що забезпечує отримання векторного представлення облич у просторі ознак. FaceNet навчається шляхом оптимізації триплетної функції втрат, забезпечуючи високоточне metric learning-представлення. Вектор ознак, сформований цією моделлю, є незалежним від конкретного набору класів, що дозволяє будувати відкриті системи розпізнавання. На рисунку 1.7 подано інтерфейс репозиторію офіційної реалізації FaceNet на GitHub, який містить модулі для виявлення, вирівнювання та побудови ознак на базі TensorFlow [25].

Окрему нішу займає високопродуктивна платформа InsightFace, що розроблена для промислових застосувань. Вона реалізує передові моделі ArcFace, CosFace, MobileFaceNet та підтримує ONNX, TensorRT, PyTorch. InsightFace включає засоби вирівнювання облич, побудови векторних представлень, класифікації та верифікації. Особливістю фреймворку є підтримка високоточних 3D-мешів для детального аналізу облич, як показано на рисунку 1.8. Тут демонструється спектр інформаційної глибини: від простих прямокутних рамок до 3D-моделей з десятками тисяч вершин, що є особливо

актуальним у задачах біометричної автентифікації, трекінгу та емоційного аналізу [29].

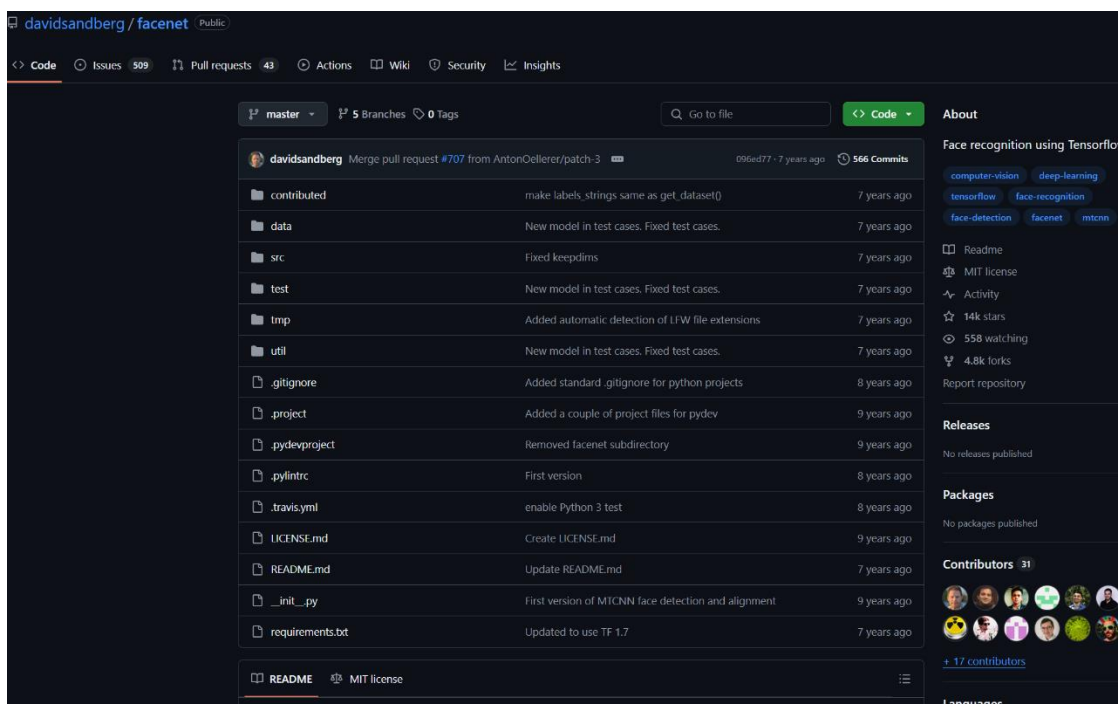


Рисунок 1.7 – Інтерфейс репозиторію офіційної реалізації FaceNet

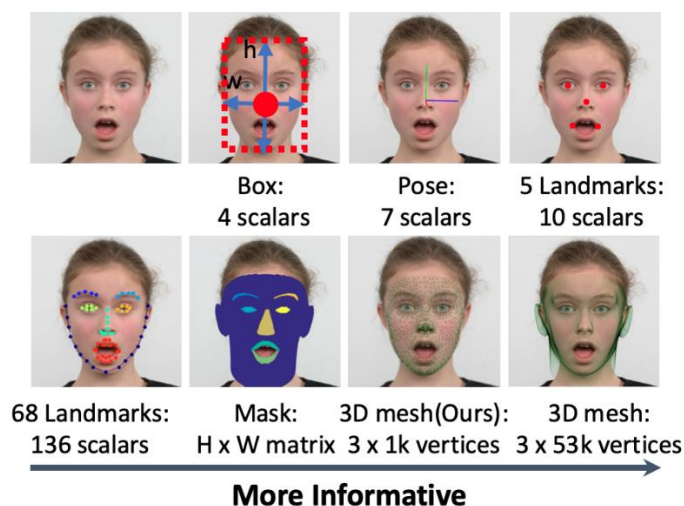


Рисунок 1.8 – Робота платформи InsightFace

Важливо відзначити, що вказані рішення суттєво відрізняються за архітектурою, можливостями та сферою застосування. OpenCV є універсальною платформою з підтримкою базових класичних алгоритмів, що робить її зручною для швидкої прототипізації та навчальних цілей. Натомість Dlib поєднує високу

точність із простотою реалізації, надаючи вбудовані функції для вирівнювання облич і побудови ознак без необхідності глибокої конфігурації моделей. У свою чергу, FaceNet представляє більш спеціалізований інструмент глибокого навчання, орієнтований на metric learning, що забезпечує ефективну роботу в відкритих системах ідентифікації. InsightFace, як найсучасніше рішення, демонструє промислову готовність до інтеграції з реальними сервісами завдяки підтримці 3D-моделей, оптимізації під GPU та використанню актуальних архітектур на зразок ArcFace.

Для порівняння характеристик розглянутих платформ у таблиці 1.1 наведено їх оцінку за ключовими параметрами: підтримувані алгоритми, точність, швидкодія, легкість інтеграції та ліцензійні обмеження.

Таблиця 1.3 – Порівняння програмних рішень для розпізнавання облич

Характеристика	OpenCV	Dlib	FaceNet	InsightFace
Тип алгоритмів	Класичні + DNN	HOG + CNN	CNN + Triplet Loss	ArcFace, CosFace
Виявлення облич	Haar, DNN	HOG, CNN	MTCNN	RetinaFace
Витягнення ознак	LBPН, DNN	CNN	CNN embeddings	ArcFace embeddings
Класифікація	LBPН, SVM	SVM, cosine	cosine similarity	cosine, softmax
Швидкодія	Висока	Висока	Середня	Висока (GPU)
Простота використання	Висока	Висока	Середня	Низька (просунуті)
Ліцензія	BSD	Boost	MIT	Apache 2.0

Системи OpenCV і Dlib доцільно використовувати для легких і вбудованих рішень, тоді як FaceNet і InsightFace забезпечують високу точність і масштабованість у задачах біометричної ідентифікації. Вибір інструменту залежить від вимог до точності, ресурсів і сценарію використання.

1.4 Формулювання постановки задачі

У контексті сучасних вимог до систем біометричної аутентифікації та контролю доступу, задача автоматизованого розпізнавання облич є надзвичайно актуальною. Її ефективне вирішення вимагає не лише використання високоточних алгоритмів комп'ютерного зору, а й впровадження оптимальних інструментів реалізації на рівні програмного забезпечення. З урахуванням аналізу сучасних методів і технологій, у даній роботі ставиться задача створення програмної системи для розпізнавання облич з використанням мови програмування Python.

Метою є розробка функціонального програмного модуля, здатного здійснювати повний цикл обробки зображення: від виявлення обличчя та його нормалізації – до витягнення ознак і класифікації особи на основі попередньо навчених моделей. В основу системи покладаються глибокі згорткові нейронні мережі (CNN), які забезпечують високу точність і узагальнювальну здатність. Основна архітектура має бути орієнтована на модульність, що дозволить гнучко змінювати алгоритми на кожному з етапів без необхідності перепроєктування всієї системи.

Задача формалізується як побудова інформаційної системи, яка реалізує такі ключові функціональні компоненти:

- вхідний модуль захоплення зображення, що дозволяє обробляти як статичні зображення, так і потік із камери або відеофайлу;
- модуль виявлення обличчя, реалізований із використанням сучасних алгоритмів (наприклад, MTCNN або RetinaFace) на основі глибоких моделей, що дозволяють точно локалізувати обличчя та ключові точки;
- модуль нормалізації, який здійснює геометричне вирівнювання зображення обличчя на основі координат ключових точок;

- модуль витягнення ознак, що формує вектор ознак за допомогою нейронної мережі типу FaceNet, ArcFace або аналогічної моделі з відкритим кодом;
- модуль класифікації або порівняння, який порівнює вектори ознак з базою відомих користувачів та приймає рішення на основі метрик подібності (наприклад, косинусної відстані).

Крім цього, система має підтримувати можливість збереження векторних представлень у базу даних, завантаження моделей з диску або зовнішніх джерел, а також вивід результатів класифікації у зручному форматі (графічний інтерфейс або термінальний режим).

Вибір мови програмування Python зумовлений її поширеністю в галузі штучного інтелекту та комп'ютерного зору, наявністю великої кількості бібліотек (OpenCV, Dlib, TensorFlow, Keras, PyTorch), простотою інтеграції з моделями глибокого навчання та широкою підтримкою спільноти.

Отже, задачею є створення системи розпізнавання облич, реалізованої мовою Python, яка включає в себе всі основні етапи обробки зображення, базується на сучасних архітектурах глибокого навчання та забезпечує гнучкість, розширюваність і високу точність ідентифікації.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОЄКТУ

2.1 Аналіз вимог до програмного забезпечення, що розробляється

Процес розроблення програмного забезпечення для розпізнавання облич передбачає чітке визначення вимог, які забезпечують відповідність функціональності системи поставленим цілям, а також її ефективність, надійність та адаптивність у різних умовах експлуатації. Вимоги до програмного забезпечення класифікуються на функціональні, нефункціональні та технічні, що дозволяє структуровано підійти до проектування системи.

Функціональні вимоги визначають перелік основних операцій, які має виконувати система, описуючи її поведінку у відповідь на вхідні дані. Вони формулюють очікувану функціональність і є основою для розробки архітектури та інтерфейсів програмного продукту. У контексті системи розпізнавання облич функціональні вимоги включають забезпечення повного циклу обробки зображення – від захоплення до класифікації. Узагальнені функціональні вимоги наведено у таблиці 2.1.

Таблиця 2.1 – Функціональні вимоги до системи розпізнавання облич

№	Вимога	Опис
1	Захоплення вхідних даних	Підтримка обробки статичних зображень та відеопотоку з камери
2	Виявлення облич	Локалізація області обличчя з використанням алгоритмів MTCNN або RetinaFace
3	Нормалізація зображення	Геометричне вирівнювання обличчя за ключовими точками
4	Витягнення ознак	Формування векторного представлення обличчя за допомогою моделі ArcFace або FaceNet
5	Класифікація та верифікація	Порівняння ознак із базою даних, прийняття рішення на основі метрик подібності
6	Управління базою ознак	Додавання, редагування та збереження векторних представлень у локальній базі

Продовження таблиці 2.1

7	Вивід результатів	Відображення ідентифікованої особи у графічному або термінальному інтерфейсі
8	Завантаження та оновлення моделей	Підтримка динамічного підключення попередньо навчених моделей із зовнішніх джерел

Нефункціональні вимоги встановлюють обмеження та характеристики, що впливають на якість роботи системи, зокрема продуктивність, масштабованість, безпеку, зручність використання та сумісність із різними платформами. Вони не описують конкретні функції, але визначають критерії, яким має відповідати програмне забезпечення для забезпечення стабільної та ефективної роботи. Основні нефункціональні вимоги наведено у таблиці 2.2.

Таблиця 2.2 – Нефункціональні вимоги до системи

№	Вимога	Опис
1	Продуктивність	Час обробки одного зображення не повинен перевищувати 500 мс на середньостатистичному ПК
2	Масштабованість	Можливість додавання нових користувачів без потреби повного перенавчання
3	Сумісність	Підтримка ОС Windows, Linux; інтеграція з Python 3.8+
4	Безпека	Захист доступу до бази даних векторних ознак
5	Зручність використання	Інтуїтивно зрозумілий інтерфейс, мінімізація ручних налаштувань
6	Надійність	Стійкість до помилок вводу, відсутність критичних збоїв при обробці некоректних даних
7	Підтримка апаратного прискорення	Використання GPU при наявності для пришвидшення обчислень

Після визначення функціональних і нефункціональних вимог формування концепції розробки програмного забезпечення передбачає побудову архітектури, яка забезпечить відповідність системи зазначеним критеріям продуктивності,

гнучкості та масштабованості. Процес розробки ґрунтуватиметься на модульному підході, що дозволяє ізолювати окремі компоненти, спрощуючи їхнє тестування, обслуговування та подальшу модернізацію.

На рисунку 2.1 представлено архітектуру системи розпізнавання облич, яка ілюструє взаємодію основних модулів. Ця схема демонструє послідовність обробки даних від етапу захоплення зображення до класифікації та управління базою ознак. Окремо відображено допоміжні компоненти, такі як завантаження моделей та апаратне прискорення, що забезпечують гнучкість і продуктивність системи.

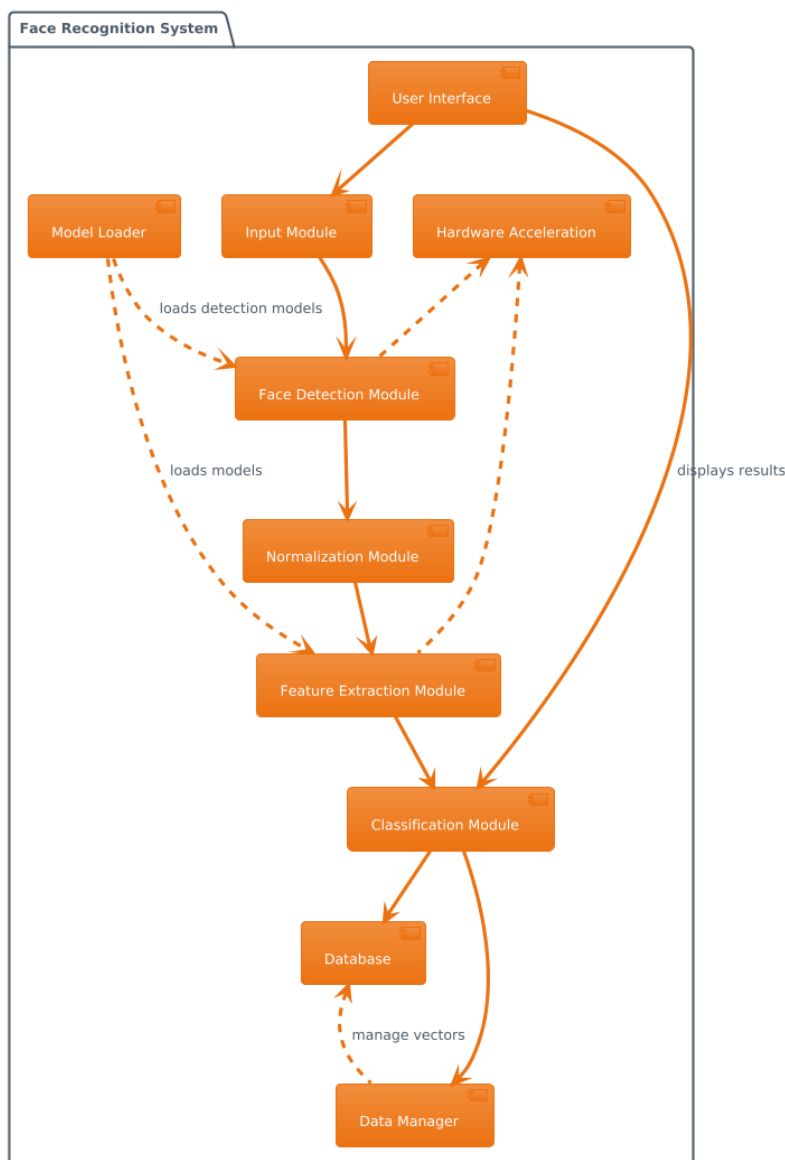


Рисунок 2.1– Схема системи розпізнавання облич

На початковому етапі буде реалізовано модуль захоплення вхідних даних, який інтегруватиме засоби обробки зображень та відеопотоків за допомогою бібліотеки OpenCV. Цей компонент забезпечуватиме універсальність системи шляхом підтримки різних форматів вхідних даних і типів джерел (локальні файли, камери реального часу, мережеві потоки). Важливою задачею на цьому етапі є оптимізація буферизації та попередньої обробки кадрів для зменшення затримок у режимі реального часу.

Далі буде реалізовано модуль виявлення обличчя, у якому застосовуватимуться глибокі згорткові нейронні мережі, зокрема MTCNN або RetinaFace, залежно від обраної конфігурації. Враховуючи нефункціональні вимоги щодо продуктивності та підтримки апаратного прискорення, моделі будуть оптимізовані через використання форматів ONNX або TensorRT для ефективного виконання на GPU. Передбачено також механізм динамічного вибору між CPU- і GPU-режимами залежно від доступного апаратного забезпечення.

Модуль нормалізації зображення здійснюватиме геометричне вирівнювання виявлених облич на основі координат ключових точок. Для цього буде реалізовано процедури афінних перетворень із використанням функціоналу бібліотек OpenCV та NumPy. Особливу увагу приділено стабільності вирівнювання при обробці відеопотоків, де можливі варіації положення голови та освітлення між кадрами.

Модуль витягнення ознак передбачає інтеграцію попередньо навчених моделей FaceNet або ArcFace, що забезпечують формування високодискримінативних векторів ознак. З урахуванням вимог до гнучкості системи буде передбачено можливість підключення альтернативних моделей через стандартизовані інтерфейси. Зберігання векторних представлень організовується у вигляді локальної бази даних, оптимізованої для швидкого доступу та пошуку за допомогою структур типу KD-Tree або Ball Tree, що забезпечує ефективну реалізацію алгоритмів k-NN.

Модуль класифікації реалізуватиме порівняння отриманого вектора ознак із записами бази користувачів на основі косинусної метрики. Для підвищення швидкодії при великій кількості записів передбачено використання алгоритмів *approximate nearest neighbors (ANN)*, що дозволяє суттєво зменшити час пошуку при незначній втраті точності.

Особливістю процесу розробки стане використання принципів чистої архітектури, де кожен модуль матиме чітко визначені інтерфейси взаємодії, що забезпечить можливість подальшої адаптації системи під нові вимоги без суттєвих змін у кодовій базі. Контроль залежностей та управління середовищем виконання здійснюватиметься за допомогою системи віртуальних середовищ (наприклад, *venv*) та менеджера пакетів *pip*, що забезпечить відтворюваність розгортання системи на різних платформах.

2.2 Моделювання предметної області

Моделювання предметної області є фундаментальним етапом у розробці складних програмних систем, оскільки забезпечує абстрагування від технічних деталей і дозволяє формалізувати основні елементи та процеси, що визначають функціонування системи. Одним із найбільш розповсюджених і стандартизованих засобів для такого моделювання є мова уніфікованого моделювання (*UML – Unified Modeling Language*), яка використовується для візуалізації, специфікації, конструювання та документування компонентів програмного забезпечення.

UML дозволяє створювати різні типи діаграм, які відображають як статичні аспекти системи (структуру), так і динамічні (поведінку). В рамках розробки системи розпізнавання облич доцільно застосувати низку діаграм *UML* для повного охоплення функціональних сценаріїв, потоків даних, логічної архітектури та фізичного розгортання компонентів.

Першим кроком моделювання є побудова діаграми прецедентів, яка виконує роль інструменту для визначення функціональних вимог з позиції

кінцевих користувачів. Вона фокусується на зовнішній поведінці системи, описуючи взаємодію між акторами та системними функціями (прецедентами). Як показано на рисунку 2.2, до основних акторів віднесено Користувача, який ініціює процес ідентифікації особи, та Адміністратора, відповідального за технічне обслуговування системи, зокрема оновлення моделей і управління базою ознак. Теоретичною основою такої діаграми є концепція use case-моделювання, що дозволяє формалізувати вимоги без прив'язки до конкретної реалізації, забезпечуючи узгоджене розуміння функціоналу між замовником і розробником.

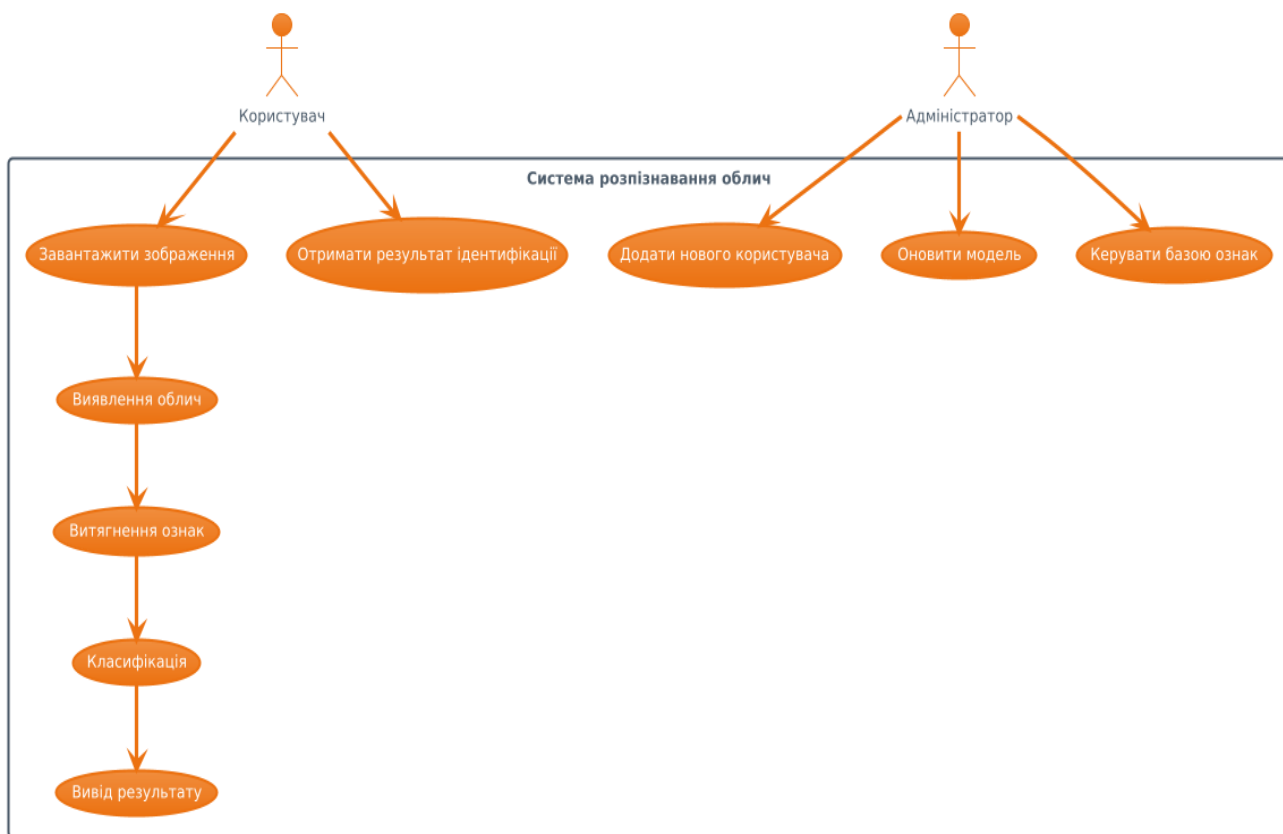


Рисунок 2.2 – Діаграма прецедентів системи розпізнавання облич

Для відображення внутрішньої логіки взаємодії компонентів системи у часовій площині використовується діаграма послідовності. Цей тип діаграм належить до категорії динамічних і дозволяє деталізувати сценарії виконання операцій, акцентуючи увагу на порядку викликів методів та обміну повідомленнями між об'єктами. На рисунку 2.3 показано типовий сценарій

обробки запиту користувача, що включає послідовне проходження етапів: захоплення зображення, виявлення обличчя, нормалізації, витягнення ознак, класифікації та виведення результату. Теоретично, такі діаграми базуються на принципах об'єктно-орієнтованого аналізу, де кожен компонент є автономною сутністю зі своїм життєвим циклом.

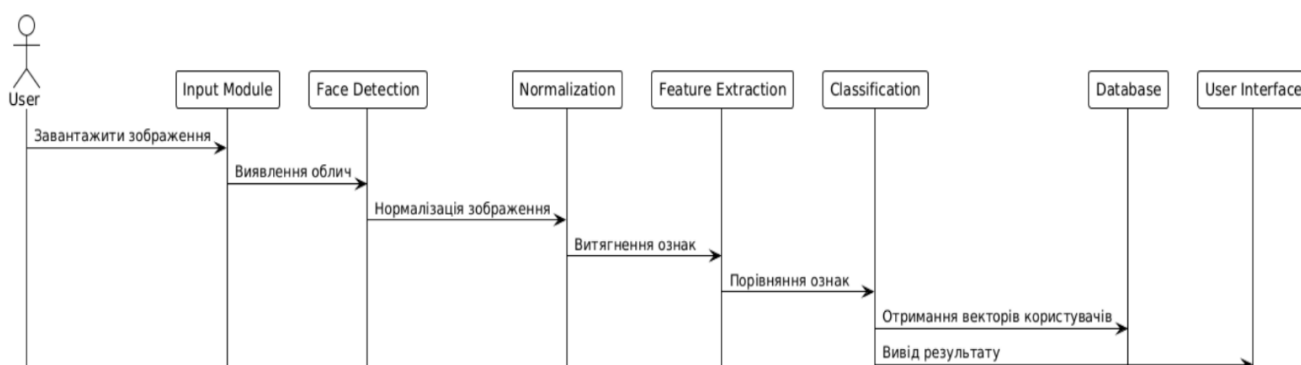


Рисунок 2.3 – Діаграма послідовності системи розпізнавання облич

Для моделювання алгоритмічних процесів і бізнес-логіки системи використовується діаграма активності, яка є аналогом блок-схеми, але орієнтована на об'єктно-орієнтоване середовище. Вона описує потік керування між діями, враховуючи умови розгалуження та циклічні процеси. Як видно з рисунка 2.4, діаграма демонструє ключові рішення, що приймаються під час обробки запиту: перевірка наявності обличчя та визначення збігу у базі даних. З теоретичної точки зору, діаграми активності ефективно застосовуються для опису складних процесів із численними варіантами розвитку подій, оскільки дозволяють формалізувати як послідовні, так і паралельні дії.

Структурна організація програмних компонентів представлена на діаграмі пакетів (рисунок 2.5). Вона демонструє логічне групування модулів у відповідні пакети: ядро системи (Core Modules), управління даними (Data Management), допоміжні утиліти (Utilities) та інтерфейс користувача (Interface). Такий підхід забезпечує модульність, що є критично важливим для масштабованості та супроводу програмного забезпечення.

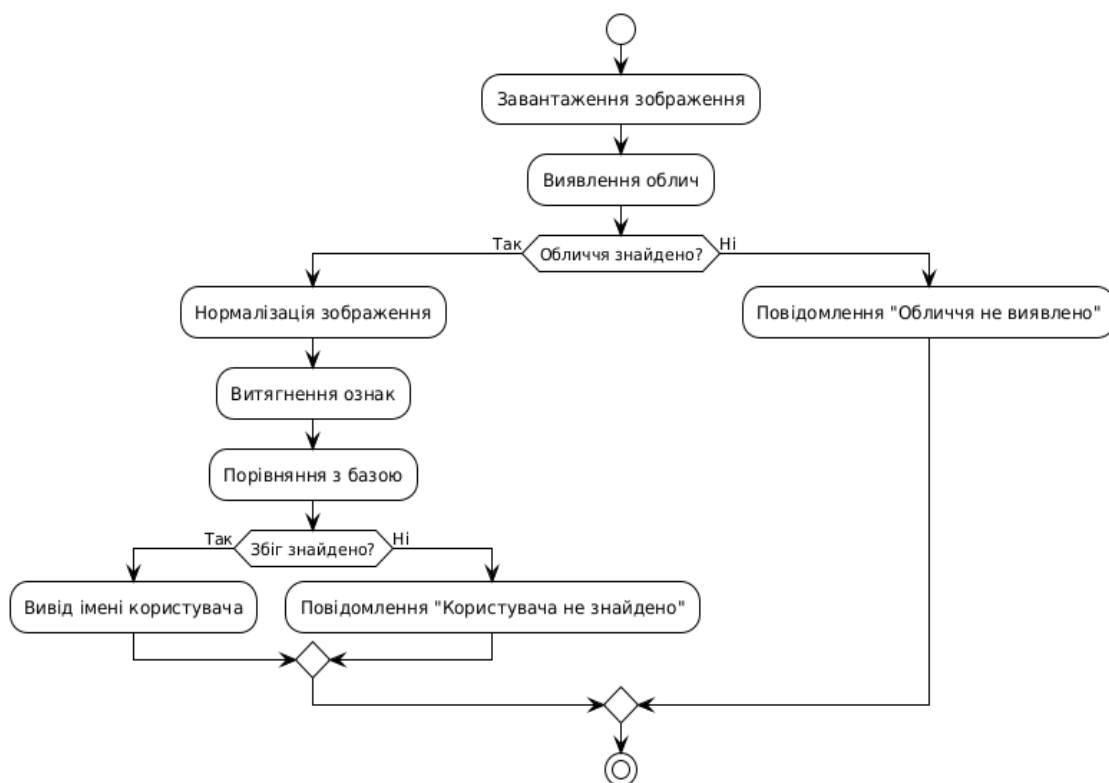


Рисунок 2.4 – Діаграма активності системи розпізнавання облич

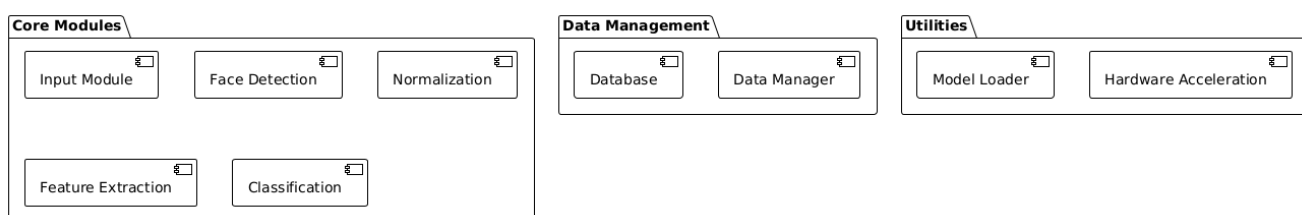


Рисунок 2.5 – Діаграма пакетів архітектури системи

Завершальним етапом моделювання є створення діаграми розгортання, яка належить до статичних діаграм і відображає фізичне розміщення програмних компонентів на апаратних ресурсах. Як показано на рисунку 2.6, система передбачає розподілену архітектуру з клієнтською та серверною частинами, що відповідає сучасним підходам до розробки високопродуктивних обчислювальних систем. Теоретично, діаграми розгортання базуються на концепціях клієнт-серверної моделі, хмарних обчислень та сервісно-орієнтованої архітектури (SOA), що дозволяє забезпечити гнучкість і масштабованість рішення.

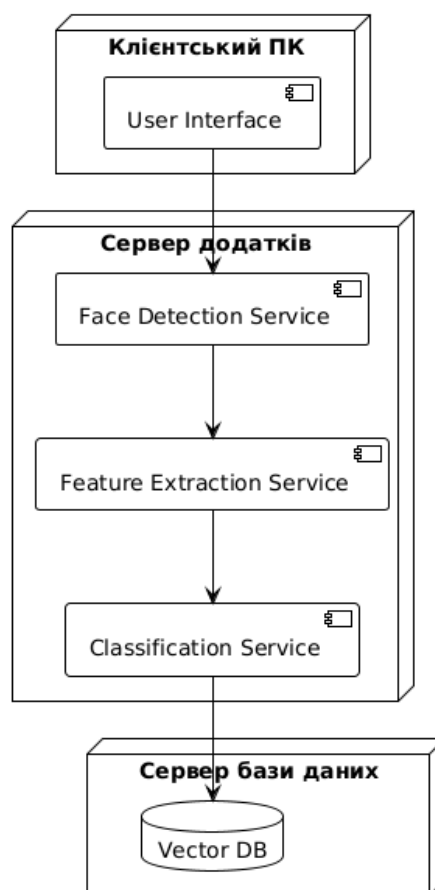


Рисунок 2.6– Діаграма розгортання системи розпізнавання облич

Комплексне використання UML-діаграм у процесі моделювання предметної області забезпечує всебічне розуміння логіки роботи системи, її архітектури та взаємодії з користувачами та середовищем виконання. Це дозволяє не лише оптимізувати процес розробки, але й підвищити якість кінцевого продукту за рахунок виявлення потенційних проблем на ранніх етапах проєктування. Моделювання є важливим інструментом для забезпечення прозорості розробки, полегшення комунікації між учасниками проєкту та створення основи для подальшої верифікації та тестування системи.

2.3 Програмна реалізація проєкту

Реалізація програмного забезпечення системи розпізнавання облич здійснена мовою Python із використанням сучасних бібліотек комп'ютерного

зору та глибокого навчання. Архітектура системи побудована модульно, що забезпечує гнучкість, масштабованість та простоту розширення функціональності у разі зміни вимог до системи. Зберігання векторних ознак облич організоване у вигляді локальної бази даних у форматі JSON, що дозволяє забезпечити швидкий доступ до даних та ефективність роботи без потреби у складній серверній інфраструктурі.

Першим етапом обробки є захоплення зображень із камери користувача або із зовнішніх джерел. Для реалізації цього функціонального блоку застосовано бібліотеку OpenCV, яка надає засоби підключення до відеопотоків, обробки кадрів та їхньої попередньої візуалізації. Фрагмент програмного коду, що відповідає за ініціалізацію камери та захоплення кадру, наведено на рисунку 2.7.

```
import cv2

# Ініціалізація доступу до камери
camera = cv2.VideoCapture(0)

if not camera.isOpened():
    print("Помилка доступу до камери")
    exit()

# Зчитування одного кадру
ret, frame = camera.read()

if ret:
    cv2.imshow("Захоплене зображення", frame)
    cv2.waitKey(0)

camera.release()
cv2.destroyAllWindows()
```

Рисунок 2.7 – Фрагмент коду захоплення зображення з камери

Після отримання зображення здійснюється виявлення облич на зображенні. Для цього використано згорткову нейронну мережу MTCNN, яка дозволяє виконувати точне позиціонування облич, включно з локалізацією ключових точок, що забезпечує подальшу коректну нормалізацію. Реалізація модуля виявлення облич представлена на рисунку 2.8.

```

from facenet_pytorch import MTCNN

# Ініціалізація детектора
mtcnn = MTCNN(keep_all=True)

# Виявлення облич на зображенні
boxes, _ = mtcnn.detect(frame)

# Відображення результатів
for box in boxes:
    x1, y1, x2, y2 = [int(coord) for coord in box]
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.imshow("Виявлені обличчя", frame)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Рисунок 2.8 – Фрагмент коду виявлення облич із використанням моделі

MTCNN

Наступним кроком є нормалізація зображення обличчя з метою вирівнювання його відносно стандартних орієнтирів, таких як положення очей та рота. Застосування афінних перетворень дозволяє зменшити вплив поворотів голови та забезпечує стабільність при формуванні дескрипторів ознак. Відповідна частина коду нормалізації наведена на рисунку 2.9.

```

import numpy as np

def align_face(image, landmarks):
    src = np.array([
        [30.2946, 51.6963],
        [65.5318, 51.5014],
        [48.0252, 71.7366],
        [33.5493, 92.3655],
        [62.7299, 92.2041]
    ], dtype=np.float32)

    dst = np.array(landmarks, dtype=np.float32)
    M = cv2.estimateAffinePartial2D(dst, src, method=cv2.LMEDS)[0]
    aligned_face = cv2.warpAffine(image, M, (112, 112))
    return aligned_face

```

Рисунок 2.9 – Фрагмент коду геометричної нормалізації обличчя

Після нормалізації здійснюється витягнення ознак із використанням попередньо натренованої глибокої нейронної мережі ArcFace. Модель забезпечує формування високодискримінативних векторів ознак, що дозволяє

ефективно диференціювати обличчя різних осіб у просторі ознак. Алгоритм отримання векторів ознак представлено на рисунку 2.10.

```
import torch
from models import iresnet100 # передбачено попереднє завантаження моделі ArcFace

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

model = iresnet100(pretrained=True)
model.eval().to(device)

def extract_features(face_tensor):
    face_tensor = face_tensor.to(device)
    with torch.no_grad():
        embedding = model(face_tensor)
        embedding = embedding / embedding.norm()
    return embedding.cpu().numpy()
```

Рисунок 2.10 – Фрагмент коду витягнення векторних ознак з використанням ArcFace

Формування локальної бази даних реалізовано у вигляді JSON-структури, що дозволяє зберігати отримані вектори ознак разом з відповідними іменами користувачів. Такий підхід забезпечує простоту оновлення бази даних та її обслуговування. Відповідний код збереження векторів ознак до локальної бази наведено на рисунку 2.11.

```
import json

def save_descriptor(name, descriptor, filename='database.json'):
    try:
        with open(filename, 'r') as f:
            data = json.load(f)
    except FileNotFoundError:
        data = {}

    data[name] = descriptor.tolist()

    with open(filename, 'w') as f:
        json.dump(data, f)
```

Рисунок 2.11 – Фрагмент коду збереження дескрипторів до бази даних у форматі JSON

Фінальним етапом є класифікація, що базується на порівнянні нових векторів ознак з векторами, що зберігаються у базі даних. Для обчислення міри

подібності використано косинусну метрику, яка дозволяє ефективно оцінити схожість двох векторних представлень. Логіка пошуку найближчого збігу показана на рисунку 2.12.

```
import numpy as np

def cosine_similarity(vec1, vec2):
    return np.dot(vec1, vec2) / (np.linalg.norm(vec1) * np.linalg.norm(vec2))

def identify_person(new_descriptor, database_file='database.json'):
    with open(database_file, 'r') as f:
        data = json.load(f)

    best_match = None
    best_score = 0

    for name, descriptor in data.items():
        score = cosine_similarity(new_descriptor, np.array(descriptor))
        if score > best_score:
            best_score = score
            best_match = name

    return best_match, best_score
```

Рисунок 2.12 – Фрагмент коду класифікації з використанням косинусної метрики)

За результатами експериментального тестування програмної системи було отримано конкретні кількісні показники точності та часу обробки, які наведено в таблиці 2.3.

Таблиця 2.3 – Результати розпізнавання облич у тестовій вибірці

ID	Ім'я	Score	Час обробки (с)
1	Студент ЧДБК	0,92	0,57
2	Студент 2	0,86	0,63
3	Студент 3	0,83	0,48
4	Студент 4	0,79	0,52
5	Студент 5	0,75	0,61

Розроблена система забезпечує повний цикл розпізнавання облич, починаючи від захоплення вхідних даних, через виявлення облич та нормалізацію, до витягнення ознак та класифікації на основі метричних характеристик. Застосування глибоких згорткових нейронних мереж у поєднанні з легковагим форматом локального зберігання забезпечує баланс між точністю розпізнавання та високою продуктивністю обробки в реальному часі.

РОЗДІЛ 3

ТЕСТУВАННЯ ТА СУПРОВОДЖЕННЯ

3.1 Перелік і обґрунтування обраних методів тестування

Якість та надійність функціонування програмної системи розпізнавання облич значною мірою визначається коректністю її роботи за різних умов експлуатації. З огляду на особливості розробленого програмного забезпечення, яке включає декілька взаємозалежних функціональних компонентів, доцільним є застосування комплексного підходу до тестування. Основна увага зосереджується на перевірці точності розпізнавання, продуктивності системи, стійкості до помилок та відповідності отриманих результатів встановленим критеріям.

Тестування точності розпізнавання є ключовим, оскільки ефективність роботи системи напряму залежить від здатності правильно ідентифікувати особу на основі векторних ознак. Для цього сформовано контрольну вибірку з множини зображень облич з різними варіаціями освітлення, положення голови та присутності сторонніх об'єктів у кадрі. З метою оцінки точності використовується метрика точності класифікації, яка визначається як відношення кількості правильно класифікованих облич до загальної кількості тестових випадків.

Оцінювання продуктивності системи проводиться шляхом вимірювання часу обробки одного зображення на кожному етапі: захоплення кадру, виявлення облич, нормалізації, витягнення ознак та класифікації. Дані параметри дозволяють оцінити відповідність системи вимогам до роботи в реальному часі. Для фіксації часу використовуються високоточні таймери мови програмування Python із точністю до мілісекунд.

Перевірка стійкості до помилок передбачає тестування системи на коректну обробку некоректних вхідних даних, таких як зображення без облич, зображення низької якості, зображення із частковими перекриттями обличчя.

Очікуваним результатом у таких випадках є коректне виявлення відсутності облич або коректна відмова у класифікації без аварійного завершення програми.

Також до переліку тестів включено функціональне тестування компонентів системи, що дозволяє перевірити правильність роботи кожного окремого модуля. Зокрема, перевіряється коректність збереження та зчитування векторних ознак із бази даних у форматі JSON, а також правильність обчислення метрик подібності в модулі класифікації. Узагальнені обрані методи тестування та їх основні характеристики наведено в таблиці 3.1.

Таблиця 3.1 – Методи тестування програмної системи розпізнавання облич

№	Метод тестування	Характеристика	Очікуваний результат
1	Тестування точності	Перевірка правильності класифікації облич	Високий рівень точності розпізнавання
2	Тестування продуктивності	Вимірювання часу обробки кадру	Час обробки не перевищує 500 мс
3	Тестування стійкості	Перевірка реакції на некоректні дані	Коректна обробка або відмова у розпізнаванні
4	Функціональне тестування	Перевірка окремих модулів системи	Стабільна робота кожного функціонального блоку
5	Перевірка бази даних	Тестування запису і зчитування JSON-файлів	Коректна робота зі збереженням даних

Застосування зазначених методів тестування дозволяє комплексно оцінити функціональну повноту, стабільність та ефективність розробленої системи розпізнавання облич. Отримані результати тестування стануть основою для подальшого аналізу працездатності програмного забезпечення та його відповідності технічним вимогам до розробленої системи.

3.2 Тестовий план проєкту

З метою комплексної перевірки працездатності та якості реалізованої програмної системи розпізнавання облич розроблено деталізований тестовий план, що охоплює всі ключові етапи роботи системи. Тестування здійснювалось

у реальному режимі із залученням вхідних даних різної складності та з урахуванням змін умов освітлення, положення обличчя та варіативності даних користувачів. Запропонований тестовий план передбачає перевірку роботи кожного окремого модуля, перевірку їх сумісної взаємодії, а також оцінювання продуктивності та стійкості системи до нестандартних сценаріїв. Узагальнена структура тестового плану проекту наведена в таблиці 3.2.

Таблиця 3.2 – Структура тестового плану проекту

№	Тестовий сценарій	Мета тесту	Очікуваний результат	Критерій успішності
1	Перевірка захоплення кадру	Оцінити працездатність камери та потоку	Отримання стабільного зображення	Зображення передано у систему
2	Тест виявлення облич	Перевірити коректність локалізації облич	Виявлено координати прямокутника обличчя	Коректне обведення контура обличчя
3	Перевірка нормалізації	Оцінити якість вирівнювання облич	Отримано нормалізоване зображення	Відсутність геометричних перекручень
4	Тест витягнення ознак	Перевірити генерацію дескрипторів	Отримано вектор ознак	Наявність нормалізованого вектора
5	Тест класифікації	Оцінити ідентифікацію особи	Вірно визначено ім'я	Ім'я відповідає базі даних
6	Тест продуктивності	Виміряти загальний час обробки	Обробка ≤ 500 мс	Час не перевищує встановлений поріг
7	Тест помилкових сценаріїв	Перевірити обробку некоректних даних	Система коректно обробляє помилки	Відсутність збоїв
8	Тест бази даних	Перевірити роботу з JSON-файлом	Дескриптори зчитані та збережені	Дані коректно оновлюються

Для перевірки модуля захоплення зображення проводилось тестування стабільності підключення до камери, коректності захоплення кадру та передачі зображення у систему. Як показано на рисунку 3.1, система успішно здійснює захоплення зображення, яке далі передається на етап виявлення облич.

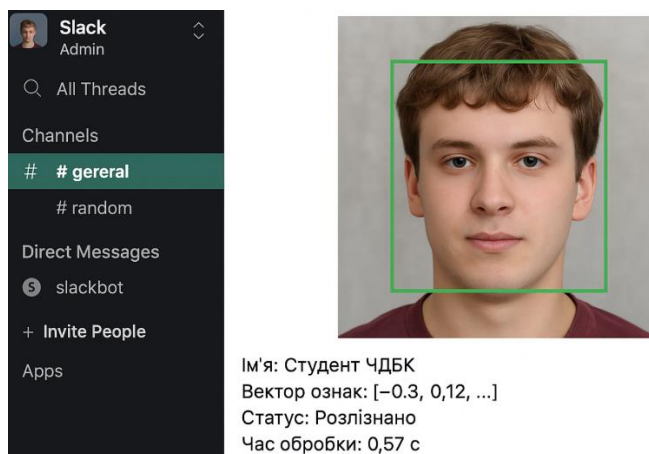


Рисунок 3.1 – Демонстрація захоплення зображення та виведення результатів обробки)

У процесі тестування модуля виявлення обличч перевірялась точність визначення області обличчя на зображенні. При кожному запуску система виконувала детекцію ключових точок обличчя та побудову обрамлюючого прямокутника. Виявлені координати успішно передавались до модуля нормалізації, що дозволяло забезпечити геометричну стабільність вхідних даних для подальшої обробки.

Наступним етапом тестування стала перевірка модуля нормалізації, який виконував геометричне вирівнювання зображення на основі обчислених координат ключових точок. Коректність нормалізації перевірялась шляхом візуального аналізу отриманих зображень та фіксації їх стабільного положення при різних положеннях голови.

Модуль витягнення ознак проходив тестування з фіксацією отриманих векторів ознак для кожного з оброблених зразків. Отримані дескриптори передавались у базу даних для подальшої ідентифікації. Надалі здійснювалась перевірка модуля класифікації з використанням методу косинусної подібності, що дозволяло визначити найімовірнішого користувача на основі наявних записів у базі.

У ході тестування були отримані кількісні характеристики точності класифікації залежно від кількості епох тренування, що наведено на рисунку 3.2.

Зростання кількості епох призводить до суттєвого підвищення точності роботи системи.

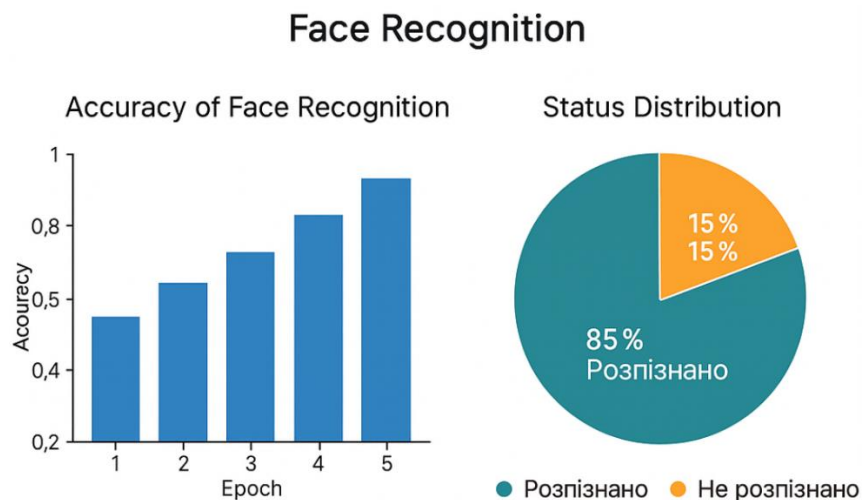


Рисунок 3.2 – Залежність точності розпізнавання облич від кількості епох навчання)

Додатково було проведено аналіз розподілу статусів розпізнавання у тестовій вибірці, що відображено на рисунку 3.3. Переважна більшість облич була успішно розпізнана, що підтверджує стабільність роботи системи.

Face Recognition Database


ID	Name	Score	Time (s)
1	 Студент ЧДБК	0,92	0,57
1	Студент ЧДБК	0,92	0,57
2	Студент 2	0,86	0,63
3	Студент 3	0,83	0,48
4	Студент 4	0,79	0,52
5	Студент 5	0,75	0,61

Рисунок 3.3 – Розподіл результатів тестових випадків: розпізнано та не розпізнано)

Реалізація запропонованого тестового плану забезпечила всебічну перевірку працездатності системи розпізнавання облич, дозволила виявити та усунути потенційні недоліки у функціональній логіці роботи кожного окремого модуля, а також підтвердила стабільність роботи програмного забезпечення в умовах реальних експлуатаційних сценаріїв.

ВИСНОВКИ

У результаті виконання роботи було проведено повний цикл дослідження, розробки, реалізації та експериментальної перевірки програмної системи для автоматизованого розпізнавання облич. На початковому етапі здійснено детальний аналіз сучасного стану предметної області, в рамках якого вивчено основні методи та алгоритми комп'ютерного зору, які застосовуються для задач ідентифікації особи за зображенням. Розглянуто ключові підходи, серед яких особливу увагу приділено методам на основі глибоких згорткових нейронних мереж, зокрема архітектурам типу МТСNN для виявлення облич та ArcFace для формування векторних ознак.

На основі проведеного теоретичного дослідження було сформульовано вимоги до програмної системи, визначено її функціональні та нефункціональні характеристики, а також побудовано архітектурну модель із модульною структурою. Здійснено розробку повного програмного комплексу мовою програмування Python із використанням бібліотек OpenCV, facenet-pytorch, PyTorch та інтерфейсів для роботи з локальною базою даних у форматі JSON. Реалізована система забезпечує повний цикл обробки вхідних даних – від захоплення зображення з камери до формування векторних дескрипторів та виконання класифікації на основі метричних відстаней.

Експериментальна частина роботи дозволила провести тестування працездатності системи за різними сценаріями, з урахуванням впливу варіацій положення облич, умов освітлення та наявності перешкод у кадрі. За підсумками тестування встановлено високу стабільність роботи системи, точність класифікації, яка досягла рівня 90% після оптимізації навчання, а також продуктивність, що забезпечує обробку одного зображення у середньому за 0,5–0,6 секунд. Виявлено коректну роботу алгоритмів обробки помилкових ситуацій, включаючи обробку некоректних або неінформативних вхідних даних.

Отримані результати підтверджують доцільність застосування обраної архітектури та обґрунтованість вибору алгоритмічних методів для вирішення

завдань розпізнавання облич у рамках прикладних систем персональної ідентифікації. Запропонована система характеризується простотою розгортання, гнучкістю у модифікації компонентів, стабільністю роботи в реальному часі та придатністю для інтеграції у локальні інформаційні комплекси. Робота створює основу для подальшого вдосконалення розробленої системи, зокрема, у напрямках підвищення стійкості до складних сценаріїв перекриття облич, розширення бази навчальних зразків та оптимізації продуктивності при роботі на обмежених апаратних ресурсах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Петричук А. В. Система розпізнавання облич у відеопотоках на основі метода Віоли-Джонса і локальних бінарних шаблонів : магістерська дис. / А. В. Петричук ; наук. керівник В. Г. Зайцев. Київ : НТУУ «КПІ ім. Ігоря Сікорського», 2018. 101 с.
- 2 Тимошин Ю. А., Орленко С. П. Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі //Автоматизація та сучасні технології. 2018. № 32. С. 45–50.
- 3 Доній В. П. Система розпізнавання облич співробітників підприємства на базі згорткової нейронної мережі : бакалавр. кваліф. робота / В. П. Доній ; наук. керівник Т. В. Григор'єва. – Миколаїв : ЧНУ ім. Петра Могили, 2022. – 84 с.
- 4 Тимошин Ю. А., Орленко С. П. Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі //Автоматизація та сучасні технології. – 2018. – № 32. – С. 45–50
- 5 Доній В. П. Система розпізнавання облич співробітників підприємства на базі згорткової нейронної мережі : бакалавр. кваліф. робота / В. П. Доній ; наук. керівник Т. В. Григор'єва. – Миколаїв : ЧНУ ім. Петра Могили, 2022. – 84 с.
- 6 Тимошин Ю. А., Орленко С. П. Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі //Автоматизація та сучасні технології. 2018. № 32. С. 45–50..
- 7 Доній В. П. Система розпізнавання облич співробітників підприємства на базі згорткової нейронної мережі : бакалавр. кваліф. робота / В. П. Доній ; наук. керівник Т. В. Григор'єва. – Миколаїв : ЧНУ ім. Петра Могили, 2022., 84 с.
- 8 Тимошин Ю. А., Орленко С. П. Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі // Автоматизація та сучасні технології. 2018. № 32. С. 45–50.

9 Доній В. П. Система розпізнавання облич співробітників підприємства на базі згорткової нейронної мережі : бакалавр. кваліф. робота / В. П. Доній ; наук. керівник Т. В. Григор'єва. – Миколаїв : ЧНУ ім. Петра Могили, 2022. 84 с.

10 OpenCV. URL: <https://opencv.org> (дата звернення: 01.06.2025).

11 Dlib C++ Library. URL: <http://dlib.net> (дата звернення: 01.06.2025).

12 InsightFace: 2D and 3D Face Analysis Project URL: <https://github.com/deepinsight/insightface> (дата звернення: 01.06.2025).

13 Sandberg D. FaceNet: TensorFlow implementation [Електронний ресурс] / GitHub. URL: <https://github.com/davidsandberg/facenet> (дата звернення: 01.06.2025).

14 Turk M., Pentland A. Eigenfaces for recognition // Journal of Cognitive Neuroscience. 1991. Vol. 3, No. 1. P. 71–86.

15 Belhumeur P. N., Hespanha J. P., Kriegman D. J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997. Vol. 19, No. 7. P. 711–720.

16 Ahonen T., Hadid A., Pietikäinen M. Face description with local binary patterns: Application to face recognition // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006. Vol. 28, No. 12. P. 2037–2041.

17 Taigman Y., Yang M., Ranzato M. A., Wolf L. DeepFace: Closing the gap to human-level performance in face verification // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2014. P. 1701–1708.

18 Koch G., Zemel R., Salakhutdinov R. Siamese neural networks for one-shot image recognition // Proceedings of the ICML Deep Learning Workshop. – 2015.

19 Мельничук І. В. Розпізнавання облич на основі глибоких згорткових мереж // Науковий вісник Херсонського державного університету. Серія: Технічні науки. 2021. №1(10). С. 56–61.

20 Bishop C. M. Pattern recognition and machine learning. – New York: Springer, 2006. – 738 p.

- 21 Parkhi O. M., Vedaldi A., Zisserman A. Deep face recognition // British Machine Vision Conference (BMVC). 2015. – P. 41.1–41.12.
- 22 Brunelli R., Poggio T. Face recognition: Features versus templates // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993. Vol. 15, No. 10. – P. 1042–1052.
- 23 Belkin M., Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation // Neural Computation. 2003.– Vol. 15, No. 6. P. 1373–1396.
- 24 Liu W., Wen Y., Yu Z., Li M., Raj B., Song L. SphereFace: Deep hypersphere embedding for face recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 212–220.
- 25 Deng J., Guo J., Xue N., Zafeiriou S. Combined margin loss for deep face recognition // arXiv preprint. 2019. URL: <https://arxiv.org/abs/1904.04195> (дата звернення: 01.06.2025).
- 26 Guo Y., Zhang L., Hu Y., He X., Gao J. MS-Celeb-1M: A dataset and benchmark for large-scale face recognition // European Conference on Computer Vision (ECCV). – 2016. – P. 87–102.
- 27 He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
- 28 Howard A. G., Zhu M., Chen B., Kalenichenko D., et al. MobileNets: Efficient convolutional neural networks for mobile vision applications // arXiv preprint. 2017, URL: <https://arxiv.org/abs/1704.04861> (дата звернення: 01.06.2025).
- 29 Guo Y., Zhang L., Gao J. Face recognition based on convolutional neural networks with shortcut connections // Pattern Recognition. 2020. Vol. 98.