

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет (ННІ) Інформаційних технологій**

**УДК 004.75**

**ПОГОДЖЕНО**

**Декан факультету  
Інформаційних технологій**

Болбот І.М., д.т.н, проф.

підпис

ПІБ, вчене звання і ступінь

«\_\_» \_\_\_\_\_ 2024 р.

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**

**Завідувач кафедри  
Комп'ютерних систем, мереж та кібербезпеки**

Касаткін Д.Ю., к. пед.н, доц.

підпис

ПІБ, вчене звання і ступінь

«\_\_» \_\_\_\_\_ 2024 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**На тему:** «Розробка та дослідження комп'ютерної системи розпізнавання обличчя з використанням нейронної мережі»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: Комп'ютерні системи і мережі

Орієнтація освітньої програми: Освітньо-професійна

**Гарант освітньої програми**

Кандидат технічних наук, доцент

(науковий ступінь та вчене звання)

\_\_\_\_\_

(підпис)

Шкарупило Вадим Вікторович

(ПІБ)

**Керівник дипломного проекту**

Доктор технічних наук, професор

(науковий ступінь та вчене звання)

\_\_\_\_\_

(підпис)

Лахно Валерій Анатолійович

(ПІБ)

**Виконав**

\_\_\_\_\_

(підпис)

Комісаренко Денис Сергійович

(ПІБ студента)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«ЗАТВЕРДЖУЮ»

завідувач кафедри

комп'ютерних систем, мереж та кібербезпеки

\_\_\_\_\_ / Касаткін Д.Ю., к.п.н., доцент. /

підпис

ПІБ, вчене звання і ступінь

«\_\_» \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ

### ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Комісаренку Денису Сергійовичу

(прізвище, ім'я, по батькові)

Спеціальність (напрямок підготовки): 123 «Комп'ютерна інженерія».

Освітня програма: Комп'ютерні системи та мережі

Тема магістерської роботи: «Розробка та дослідження комп'ютерної системи розпізнавання обличчя з використанням нейронної мережі»

затверджена наказом ректора НУБІП України від « 1 » листопада 2023 р. № 1999 'С'

Термін подання завершеної роботи на кафедру 27 листопада 2024 р.

Вихідні дані до магістерської роботи: комп'ютер зі встановленим програмним забезпеченням PyCharm, встановленим Python 3.7 та імпортованою бібліотекою OpenCV, вебкамера

Перелік питань, що підлягають дослідженню:

1. Аналіз предметної області для дослідження методів розпізнавання обличчя з використанням нейронних мереж
2. Проектування системи розпізнавання обличчя
3. Побудова моделі розпізнавання обличчя з використанням штучного інтелекту

Дата видачі завдання «2» листопада 2023р.

Керівник магістерської роботи \_\_\_\_\_ / Лахно В.А. д.т.н., професор /

(підпис)

(ПІБ, вчене звання і ступінь)

Завдання прийняв до виконання \_\_\_\_\_ / Комісаренко Д. С. /

(підпис)

(ПІБ)



## РЕФЕРАТ

Пояснювальна записка: 89 с., 38 рис., 25 використаних джерел

КОМП'ЮТЕРНА СИСТЕМА, ВІДЕО, НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, АЛГОРИТМИ, LBSH, PYTHON, PUSCHARM, OPENCV.

Мета роботи – проектування та реалізація комп'ютерної системи розпізнавання обличчя людини з відео в реальному часі за допомогою нейронної мережі.

Об'єкт - комп'ютерна система, побудована з використанням нейронної мережі для ідентифікації людини

Предмет - методи та алгоритми розпізнавання обличчя з використанням штучного інтелекту

Робота складається з чотирьох розділів

У першому розділі проведено аналіз предметної області, розглянуті різні методи розпізнавання обличчя людини, їх переваги та недоліки. Також висвітлено тему нейронних мереж і принципи їх роботи

У другому розділі описано етапи проектування системи та вибір методу реалізації.

У третьому розділі детально описаний процес створення та налаштування комп'ютерної системи, аргументовано вибір інструментів для реалізації програми, детально описано код і принцип його роботи.

В результаті виконання магістерської роботи проведено аналіз, дослідження та моделювання системи розпізнавання обличчя з використанням штучного інтелекту, проведено тестування та розроблені рекомендації щодо її впровадження.

# ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Огляд та аналіз методів знаходження облич.....	12
1.2 Огляд та аналіз методів визначення облич.....	25
1.2.1 Методи, засновані на значеннях яскравості пікселів.....	26
1.2.2 Методи, засновані на характерних точках.....	29
1.3 Що таке нейронні мережі.....	31
1.4 Постановка мети й завдань дослідження.....	41
1.5 Висновки по розділу 1.....	42
2 ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ.....	43
2.1 Виявлення облич методом Віоли-Джонса.....	43
2.1.1 Розпізнавання облич методом Віоли-Джонса.....	43
2.1.2 Ознаки Хаара.....	44
2.1.3 Каскадна класифікація.....	46
2.2 Фільтр Гауса.....	47
2.3 Перетворення LBSH.....	49
2.3.1 Класичний метод.....	49
2.3.2. Рівномірні ЛБШ.....	51
2.3.3 Центральні-симетричні ЛБШ.....	52
2.4 Маска значимих областей зображення.....	53
2.5 Метод найближчого сусіда.....	55
2.6 Планування експериментальних досліджень.....	56
2.7 Висновки по розділу 2.....	58
3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ.....	50
3.1 Інструментарій для розробки програмного забезпечення .....	60
3.2 Структура проекту .....	62
3.3 Datacollect.py .....	64
3.4 Trainingdemo.py .....	66

3.5 Testmodel .....	70
3.6 Опис інтерфейсу комп'ютерної системи .....	73
3.7 Висновки до розділу 3 .....	74
4 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ.....	75
4.1 Збір даних .....	75
4.2 Тренування .....	77
4.3 Тестуванні комп'ютерної системи.....	79
4.4 Висновки до розділу 4.....	83
ВИСНОВКИ.....	84
ПЕРЕЛІК ПОСИЛАНЬ.....	86

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AAM (Active Appearance Models) – активні моделі зовнішності.

ASM (Active Shape Models) – моделі активної форми.

SNoW (Sparse Network of Windows) – розріджена мережа Вінова

LBPH (Local Binary Patterns Histogram) – це алгоритм розпізнавання обличчя, який використовується для розпізнавання обличчя людини.

BNM (Backpropagation Neural Network) – багатошарова нейронна мережа

SVM (Support Vector Machine) – метод опорних векторів

ЛБШ – локальні бінарні шаблони.

ШІ – штучний інтелект

НМ – Нейронна мережа

ID (Identity Document) – унікальний ідентифікатор

GPU (Graphics Processing Unit) – графічний процесор

IDE (Integrated Development Environment) – це інтегроване, єдине середовище розробки, яке використовується розробниками

OpenCV (Open Source Computer Vision Library) – бібліотека комп'ютерного зору з відкритим кодом

Unknown – неопізнана людина

## ВСТУП

Останнім часом штучний інтелект буквально захоплює світ, впевнено й невпинно проникаючи у всі можливі сфери людської діяльності. Його використання стало настільки повсюдним, що значущість цієї технології складно переоцінити. Однією з найбільш помітних областей застосування штучного інтелекту можна назвати відеоаналітику. Відеоаналітика - передова технологія, яка, завдяки алгоритмам комп'ютерного зору, дозволяє автоматично збирати і аналізувати безліч даних, одержаних із відеокамер, причому в режимі реального часу або з раніше записаного матеріалу.

Відеоаналітика — це не просто набір алгоритмів і кодів, а потужний інструмент, який знайшов своє застосування в охоронних системах, транспорті, торгівлі, а також у системах відеоспостереження. Однак її справжня потужність розкривається в здатності вирішувати одне з ключових завдань сучасності — розпізнавання облич на відеозображенні. Це не просто технічне питання, а важливий крок у розвитку систем контролю доступу та ідентифікації особи, що відкриває нові горизонти для безпеки та управління даними.

Попит на біометричні технології неухильно зростає, і цьому є вагоме пояснення. У різноманітних соціальних і комерційних сферах зростає потреба у надійних рішеннях для ідентифікації особи, а можливості штучного інтелекту в цьому контексті виглядають більш ніж перспективно. Наприклад, такі гіганти, як Apple, впроваджують функцію розпізнавання облич у свої технології, зокрема в FaceID. Однак застосування цієї технології не обмежується лише смартфонами; сьогодні вона стає критичним елементом під час розслідувань злочинів, автоматизованого стеження за підозрюваними, пошуку зниклих осіб, а також відіграє важливу роль у банківському секторі, зокрема в банкоматах.

Крім того, ідентифікація облич широко використовується в сертифікованих системах для перевірки особистості, таких як паспорти,

водійські посвідчення та імміграційні картки, а також у сфері інформаційної безпеки — для контролю доступу до комп'ютерів, програм або баз даних. Сучасні системи розпізнавання здатні оперативнo і точно відрізнити одну особу від іншої, використовуючи біометричні характеристики, які неможливо втратити або забути, на відміну від звичайних паролів чи ключів.

Таким чином, штучний інтелект в ролі відеоаналітики вже сьогодні змінює наш світ, надаючи нові можливості для безпеки, автоматизації та управління в реальному часі, де кожна секунда може мати значення.

Базові методи ідентифікації користувача такі як: введення пароля, використання ключа чи пред'явлення картки, які так легко загубити, забути або й взагалі, викрасти, уже здаються дещо архаїчними. Натомість біометричні системи пропонують новий рівень безпеки, спираючись на неповторні характеристики кожної людини – особливості, які неможливо підробити чи втратити як ключ чи пароль. Однак, навіть після кількох десятиліть досліджень у сфері, поки що не винайшли універсального способу, що гарантовано точно розпізнає обличчя за будь-яких умов, без винятку.

Велику перспективу зараз представляє ідентифікація людини за обличчям, що не потребує спец. обладнання або якихось складних процедур. Власне, для цього цілком може вистачити звичайної веб-камери, хоча її однієї буде явно замало. Справжньою роботою займається програма, яка аналізує кожний піксель, кожну тінь, виявляючи унікальні риси особи і приймаючи рішення миттєво, у відповідності до закладених алгоритмів.

Тут цікаво відзначити, що загальний принцип роботи таких систем може бути схожим, але деталі суттєво варіюються. Сьогодні доступні численні методи, кожен із яких пропонує свою комбінацію підходів. Від класичних рішень, таких як метод Віюлі-Джонса чи метод головних компонент, до більш просунутих варіантів, що використовують нейронні мережі та алгоритми штучного інтелекту. Геометричні підходи, пружні графи, алгоритми, що вчаться – все це утворює цілий світ можливостей.

Тим не менш, кожен метод має свої сильні та слабкі сторони. Часом вони виявляються у критичний момент, коли несподівано система "пропускає" чужу особу або ж не дозволяє законному власникові увійти. Причини можуть бути найрізноманітнішими – від відстані до камери чи кута освітлення до особистих змін, як-от нова борода чи зміни виразу обличчя. Навіть стан здоров'я може мати значення. Отже, розробники продовжують шукати ідеальні рішення, прагнучи мінімізувати затримки і підвищити точність роботи систем.

Ця робота представляє дослідження процесів розпізнавання облич за нейронних мереж, зосереджуючись на різних методах ідентифікації на відео.

Об'єкт дослідження – процес розпізнавання обличчя з використанням нейронних мереж.

Предмет дослідження – методи визначення обличчя для ідентифікації людини у відеопотоці.

Мета – розкрити та вдосконалити можливості цих технологій, подолати існуючі перешкоди й зробити розпізнавання максимально ефективно навіть у складних умовах.

Щоб досягти поставленої мети, необхідно виконати низку ключових завдань, кожне з яких вимагає ретельного підходу:

- Провести глибокий аналіз інтелектуальних технологій, застосовуваних для відеоаналізу, занурившись у деталі алгоритмів і принципів їхньої роботи;
- Здійснити експериментальне дослідження різноманітних методів розпізнавання облич, прагнучи розкрити їхню ефективність у завданнях ідентифікації особи на відеозображеннях;
- Ретельно опрацювати отримані дані, оцінити результати досліджень та виявити закономірності, що визначають успіх методів у вирішенні задач розпізнавання облич.

Наукова новизна полягає в удосконаленні підходів до розпізнавання облич, яке досягається завдяки інтеграції різноманітних методів для аналізу відеопотоку в режимі реального часу. Такий підхід не лише покращує точність ідентифікації

осіб, а й дозволяє значно скоротити обсяги використання пам'яті в процесі обробки, що критично важливо для забезпечення швидкодії системи.

Практична цінність цього дослідження не обмежується сухою теорією чи абстрактними концепціями. Ми створили дієву систему, яка здатна впевнено ідентифікувати обличчя, відкриваючи нові горизонти для використання в реальних умовах. Сфера її застосування – це не лише контроль доступу та верифікація особистості, а й будь-які ситуації, де швидкість реакції, непохитна надійність і гарантована безпека є ключовими вимогами. Ця технологія здатна кардинально змінити правила гри, пропонуючи універсальні рішення для найрізноманітніших завдань, від комерційних до стратегічних.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Використання передових технологій аналізу відео, що спираються на інтелектуальні методи (IVT), докорінно змінює підхід до обробки відеоданих, пропонуючи неперевершену швидкість і якість. Ця технологія розвивається в двох ключових напрямках – ідентифікація та відстеження. Алгоритми відеоаналізу не просто доповнюють, а фактично формують основу сучасних систем відеоспостереження, адаптуючись до заданих правил та умов.

Відстеження – це складний процес, що включає виявлення руху в кадрі, класифікацію об'єктів, які рухаються, та визначення їхніх характеристик, таких як розмір, швидкість, колір. Існує безліч варіацій методів стеження (відеодетекторів), що можуть бути застосовані в залежності від вимог завдання.

Ідентифікація – це щось більше, ніж просто розпізнавання зображень у відеозаписі. Вона включає порівняння та класифікацію знайдених об'єктів з еталонними зразками, відносить їх до певних класів чи шаблонів. Одними з найпоширеніших методів є розпізнавання обличчя та ідентифікація номерних знаків.

### **1.1 Огляд та аналіз методів визначення обличчя**

Процес розпізнавання обличчя у відеопотоці умовно можна поділити на два основні етапи. Спочатку система визначає, чи є обличчя у кожному кадрі, а вже на другому етапі – виконує розпізнавання знайдених облич, встановлюючи їхню ідентичність. Ця технологія стає дедалі досконалішою, що дозволяє отримувати точні результати навіть в умовах, які раніше вважалися проблематичними.

Визначення обличчя – це не просто пошук людської подоби серед пікселів, а складний процес встановлення правил і критеріїв, які дозволяють виділити обличчя серед безлічі інших об'єктів. Зазвичай, для цього враховується симетрія обличчя, розташування та контрастність рис (очей, носа, рота), а також текстура шкіри. Але це тільки частина картини. Розпізнавання може також включати перевірку на відповідність попередньо створеним шаблонам – своєрідним «еталонам», які визначають, чи є дана область зображення обличчям.

Серед методів визначення обличчя існує кілька технік, які заслуговують на окрему увагу. Давайте розглянемо їх сильні та слабкі сторони.

Метод Віолі-Джонса, який з'явився завдяки Полу Віолі та Майклу Джонсу у 2001 році, став першим, що показав високу ефективність в обробці зображень у режимі реального часу. Принципи, на яких базується цей підхід, є багатогранними. Перше – це представлення зображення як інтегральної матриці, де кожен елемент містить суму інтенсивностей пікселів, що лежать ліворуч і вище. Така техніка дозволяє прискорити обчислення, необхідні для виділення обличчя.

Далі використовуються так звані ознаки, схожі на знаки Хаара, які допомагають виявити характерні риси об'єкта – наприклад, зміну інтенсивності між ділянками очей та носа. Для виділення релевантних ознак застосовується метод бустінгу – методика, яка поетапно будує композицію алгоритмів машинного навчання, з кожним кроком намагаючись компенсувати помилки попередніх. У розпізнаванні обличчя бустінг дозволяє комбінувати та відбирати саме ті ознаки, які краще за все ідентифікують риси.

Ключову роль також відіграє класифікатор – функція, яка приймає рішення, чи присутнє обличчя на зображенні, і видає відповідь у вигляді "істинно" або "хибно". Це рішення ґрунтується на наборі ознак, які були обрані раніше.

Але на цьому процес не закінчується. Для покращення точності використовуються так звані каскади ознак – послідовність класифікаторів, де кожен наступний намагається виправити помилки попередніх, дозволяючи

швидко відкидати негативні області. Така структура значно пришвидшує процес, зменшуючи час обробки і підвищуючи ймовірність точного розпізнавання облич навіть у складних умовах.

Зображення, які не мають облич, відсіюються на ранніх етапах, що дозволяє значно скоротити кількість регіонів, які потребують подальшого детального аналізу. Для цієї мети використовується віконний метод сканування – техніка, де вікно пошуку переміщується по всьому зображенню, а на кожній його позиції працює класифікатор, що визначає, чи є там обличчя.

Серед переваг методу Віюли-Джонса можна виділити декілька ключових аспектів:

- Широке визнання та застосування: Це один із найвідоміших та найчастіше застосовуваних методів для розпізнавання облич, що закріпив свою популярність завдяки ефективності.
- Швидкість обробки: Використання каскадних класифікаторів дозволяє методу блискавично визначати обличчя на зображенні, що робить його ідеальним для роботи в реальному часі, коли кожна мілісекунда має значення.
- Точність і мінімізація помилкових тривог: Порівняно з іншими, менш швидкими підходами, метод демонструє відмінну точність у виявленні облич, при цьому утримуючи низький рівень помилкових спрацьовувань.

Однак, метод Віюли-Джонса має й деякі недоліки, які можуть обмежувати його застосування:

- Необхідність великої навчальної вибірки та часу на підготовку: Щоб досягти високої точності, потрібно використовувати велику базу даних із різноманітними зображеннями облич, що потребує значних обчислювальних ресурсів та тривалого навчання.
- Залежність від положення обличчя: Найкращі результати метод демонструє тоді, коли обличчя розташоване фронтально і має

звичні анатомічні риси. Тож точність може падати, коли обличчя повернуте, нахилене або має нестандартну форму.

Цей метод поєднує силу математичних алгоритмів і практичну цінність, будуючи каскади класифікаторів, які крок за кроком відсіюють непридатні регіони зображення, залишаючи тільки ті, що містять потенційні обличчя. Сканування вікном – це наче ретельне прочісування зображення, де кожна позиція ретельно перевіряється, щоб визначити наявність характерних рис.

Адаптивний бустинг, відомий як AdaBoost, – це алгоритм машинного навчання, що відноситься до ансамблевих методів і здобув популярність завдяки своїй винятковій ефективності. Винахід Йоава Фройнда та Роберта Шапіра 1996 року став революційним проривом у світі штучного інтелекту, адже з того часу він перетворився на одну з найбільш потужних і часто застосовуваних технік бустингу. Сутність його адаптивності полягає в тому, що кожен новий комітет класифікаторів "вчиться" на помилках попередніх, концентруючись на прикладах, які попередні ітерації не змогли правильно класифікувати.

Чому це працює? Бустинг загалом – це методика, яка об'єднує кілька слабких класифікаторів, начебто невеличких дерев рішень чи лінійних моделей, і на основі їхньої комбінації створює сильний прогнозуючий механізм. У випадку AdaBoost його "секретною зброєю" є здатність ітеративно коригувати ваги навчальних прикладів. Спершу всі зразки мають однакову вагу, але з кожним новим циклом процесу акцент переходить на ті приклади, які виявилися "складними" для попередніх класифікаторів. Це ніби учень, який, помітивши, що певні завдання йому не вдаються, приділяє їм більше уваги на наступному уроці.

Однією з найпоширеніших сфер використання AdaBoost є його інтеграція з іншими класифікаційними методами для підвищення їхньої ефективності, як от у вже згаданому методі Віоли-Джонса для розпізнавання облич. Саме тут алгоритм демонструє свою виняткову гнучкість: слабкі класифікатори поступово посилюють один одного, поки не формують справжній ансамбль "розумних" рішень, що значно покращує результати. Коротше кажучи, AdaBoost діє за принципом зворотного зв'язку – кожна нова ітерація вчиться на помилках

попередньої. Цей цикл продовжується, поки кілька слабких моделей, мов би розрізнені ноти, не зливаються в симфонію точних прогнозів, що забезпечує вражаючу продуктивність навіть у найскладніших завданнях машинного навчання.

Переваги AdaBoost полягають у його здатності перетворювати кілька слабких класифікаторів на потужний інструмент для складних завдань класифікації, не вимагаючи при цьому складних налаштувань. Алгоритм вміє тонко балансувати між точністю та перенавчанням, що робить його менш схильним до цієї проблеми у порівнянні з окремими слабкими моделями. Але, звісно, не обійшлося без недоліків: він вразливий до шуму та викидів, які можуть суттєво впливати на ваги, що надаються під час навчання, і, якщо використовувати надмірно складні класифікатори або багато ітерацій, ризик перенавчання знову зростає.

Одним із прикладів застосування AdaBoost є виявлення облич із використанням технології SNoW (Sparse Network of Winnows), яка спрямована на автоматичне розпізнавання та ідентифікацію облич на зображеннях або у відеопотоці. SNoW виконує цей процес у кілька етапів, кожен з яких додає специфічний "штрих" до загальної картини виявлення.

Перший етап включає підготовку зображень, де виконується шкалювання, корекція контрастності, фільтрація шуму та інші техніки для підвищення якості. Це ніби підготовка полотна перед нанесенням фарб – жодна деталь не може бути залишена без уваги, якщо хочеться досягти чіткого відтворення.

Далі вступає у гру розріджена мережа, яка складається з численних вузлів-розпізнавачів. Кожен вузол відповідає за "пильнування" конкретного аспекту обличчя, будь то очі, ніс чи губи. Вони працюють разом, як оркестр, де кожен музикант виконує свою партію, але все разом створює гармонійну мелодію.

На третьому етапі відбувається тренування розпізнавачів на базі Winnow – алгоритму, який підлаштовує ваги ознак. Це схоже на постійне налаштування музичного інструменту, щоб той звучав максимально точно. Тут позитивні зразки представляють обличчя, тоді як негативні – будь-які інші об'єкти.

Після цього, розпізнавачі SNoW застосовуються до нових зображень, працюючи паралельно, наче кілька детективів, що одночасно шукають різні підказки. Їхні висновки об'єднуються для визначення позиції обличчя на зображенні.

На завершальному етапі може бути виконана пост-обробка, яка включає корекцію положення облич, фільтрацію помилкових деталей або навіть ідентифікацію конкретних осіб. Усі ці кроки сприяють покращенню якості виявлення, особливо у складних умовах.

Таким чином, SNoW демонструє неабияку гнучкість і точність при роботі з обличчями, адже він здатен враховувати різноманітні особливості та нюанси, які звичайні алгоритми могли б проігнорувати. Це робить його ідеальним для застосування у системах відеоспостереження, автоматичного розпізнавання осіб та інших областях, де виявлення облич є критично важливим.

На виході вузлів першого рівня відбувається магія – обчислення лінійної комбінації сигналів, які генерують вхідні вузли. Ці магічні зв'язки між вузлами, втілені у вигляді ваг, слугують коефіцієнтами для цієї лінійної комбінації, допомагаючи виокремити візуальну інформацію, яка не піддається впливу освітлення, у межах локальної області зображення. Яким же чином досягається цей дивовижний ефект? За допомогою квантування, заснованого на порозі, встановленому на рівні середнього значення пікселів у цій конкретній області, що дозволяє виділити важливі деталі з хаосу [6].

Система SNoW, своєрідний шедевр в царині класифікації, була створена для тих випадків, коли кількість характеристик об'єктів, важливих для класифікації, може бути як безмежно великою, так і абсолютно непередбачуваною. Під час класифікації вхідні вузли занурюються у світ інформації, де визначається наявність певних характеристик в обробленому зображенні.

На виході знову відбувається обчислення лінійної комбінації сигналів, що з'являються на виході вхідних вузлів, за допомогою коефіцієнтів, які задаються вагами між цими вузлами. Ці обчислення сприяють виділенню зображення з

локальної області, що позбавлена впливу освітлення. Як вже зазначалося, ключовим моментом у цьому процесі є квантування зображення, яке здійснюється за допомогою порогу, що визначається як середнє значення пікселів у цій області.

SNoW спроектована таким чином, щоб бути корисною у ситуаціях, коли кількість потенційно важливих характеристик для класифікації об'єктів є величезною, але їх конкретний перелік невідомий заздалегідь. Завдяки використанню ваг між вхідними вузлами SNoW здатна встановлювати коефіцієнти лінійної комбінації, що додатково підсилює її функціональність у цій складній сфері.

Отже, ваги між вхідними вузлами – це невидимі нитки, що сплітають разом лінійні комбінації, які оживляють зображення, дозволяючи виділити найсуттєвіші деталі у величезному морі даних.

Ці перетворення виводять на поверхню компонент з локальної області зображення, що не піддається впливу освітлення. Процес квантування, що базується на середньому значенні пікселів та формують цю область, виконує свою роль з високою точністю [6]. Вразлива архітектура розрідженої мережі відкриває можливість обробки величезного спектра властивостей зображення: тут кожна незначна характеристика відсікається в процесі навчання, а це, в свою чергу, зберігає продуктивність класифікатора, не гальмуючи його функціонування [30].

Коли показник перевищує заданий поріг, автоматично ухвалюється рішення про наявність обличчя на зображенні.

Переваги:

- Ознаки, що використовуються, демонструють нечутливість до змін освітленості локальних ділянок зображення.
- Швидкість роботи вражає, адже відбувається екранування компонентів вектора ознак.
- Висока точність виявлення – це справжня сильна сторона.

Недоліки:

- Метод чутливий до шуму та викидів даних.

Нейронно-мережеві методи охоплюють цілу палітру технологій, де центральною ідеєю є послідовність перетворень сигналів через функціональні елементи – нейрони, що працюють у паралельному режимі. Головна мета навчання таких мереж полягає у зменшенні середньоквадратичної похибки. Системи для виявлення об'єктів на зображеннях, що базуються на нейронних мережах, мають ієрархічну структуру: вектор ознак спочатку проходить через грубу мережу з високим рівнем помилок другого порядку, а потім, якщо класифікація не відбулася, рішення коригується за допомогою більш точної, проте повільнішої мережі.

Переваги:

- Висока точність виявлення при належній настройці параметрів мережі.

Недоліки:

- Чутливість до шуму.
- Необхідність ретельного налаштування параметрів для досягнення задовільних результатів.
- Схильність до перенавчання.
- Висока обчислювальна складність, що веде до недостатньої швидкості обробки в реальному часі.
- Досить тривала процедура навчання.

Багатошарові нейронні мережі (БНМ)

Архітектура БНМ включає послідовно з'єднані шари, де кожен нейрон у шарі з'єднаний з усіма нейронами попереднього рівня, створюючи мережу, здатну обробляти складні взаємозв'язки між вхідними та вихідними даними. Однак, нейронна мережа з одним рівнем прийняття рішень обмежена у своїй здатності формувати лише лінійні розділові поверхні, що суттєво звужує коло задач, які вона може вирішити. Наприклад, такі мережі не здатні впоратися із

завданнями типу "виключне або", оскільки не можуть розрізняти складні та нелінійні залежності в даних.

Нейронна мережа з нелінійною функцією активації та двома рівнями прийняття рішень, навпаки, здатна формувати опуклі області в просторі рішень, розширюючи коло задач, які вона може вирішити, і дозволяючи розрізняти складні залежності. Додавання третього рівня прийняття рішень наділяє мережу можливістю формувати області будь-якої складності, включаючи невивуклі.

Для навчання багатосарових нейронних мереж використовується метод зворотного поширення помилки, що базується на градієнтному спуску у просторі ваг, з метою мінімізації загальної похибки мережі. Цей підхід дозволяє адаптувати ваги мережі на основі зіставлення очікуваних та фактичних виходів, що, в свою чергу, сприяє розвитку мережі в розв'язанні задач і узагальненні знань на нові дані. У методі зворотного поширення помилки вага коригується, рухаючись в зворотному напрямку — від виходу до входу, через численні вагові коефіцієнти, що об'єднують нейрони. Одношарова нейронна мережа, часто іменована автоасоціативною пам'яттю, здатна ефективно відновлювати зображення, які подаються на вхід. Порівняння якості реконструйованого зображення з тестовим дозволяє оцінити точність розпізнавання, що здійснює мережа.

Цей метод володіє численними позитивними характеристиками, зокрема, можливістю відновлення спотворених і зашумлених зображень. Однак для складніших завдань його застосування може бути обмеженим.

У той час як багатосарова нейронна мережа (БНМ) також виконує Пряма класифікація зображень дозволяє вирішувати більш гнучкі та складні завдання. Тому вона може відновлювати зашумлені або спотворені зображення, що є перевагою над одношаровими мережами, але з певними застереженнями щодо її ефективності для більш серйозних цілей.

У випадку прямої класифікації зображень, БНМ отримує на вхід самі зображення у необробленій формі або набір попередньо витягнутих ключових характеристик. На виході нейрон, що демонструє найвищу активність, вказує на

приналежність виявленого об'єкта до конкретного класу. Це відкриває двері до використання БНМ для класифікації зображень з вражаючою точністю.

Отже, хоча БНМ може не досягати бажаного результату в деяких складних задачах, її здатність до відновлення зображень та класифікації робить її важливим інструментом у сучасних технологіях обробки зображень.

При застосуванні прямої класифікації зображень на вхід нейронної мережі подаються необроблені зображення. У вихідному шарі, активність нейронів визначає принадлежність об'єкта до певної категорії — найбільш активний нейрон вказує на вірогідний клас розпізнаного об'єкта. Така архітектура багат шарової нейронної мережі (БНМ) відкриває широкі можливості для точного розпізнавання зображень.

Особливо корисною є здатність БНМ розпізнавати обличчя на основі таких характеристик, як відстань між очима, носом і ротом. У гібридних системах БНМ може поєднуватися з моделями Маркова для більш точного аналізу. Проте в класичному БНМ зображення зазвичай представляється у вигляді одновимірного вектора, хоча за своєю природою воно є двовимірним, що обмежує можливості мережі у врахуванні просторових зв'язків.

Згорткова нейронна мережа (CNN) та її архітектура створена для подолання цих обмежень. В ній використовуються локальні рецептивні поля, забезпечуючи локальну двовимірну пов'язаність нейронів, що дозволяє штучному інтелекту ідентифікувати ключові ознаки в будь-якій частині зображення. Застосування просторової ієрархічної організації та підвибірок сприяє покращенню стійкості до змін масштабу, зсувів, обертань та спотворень.

CNN має багат шарову архітектуру, в якій кожен шар складається з декількох площин. Нейрони у кожній площині наступного шару пов'язані лише з невеликою кількістю нейронів попереднього шару в межах локальної області, а їхні ваги є спільними по всій площині, що реалізується через згорткові шари. Після кожного згорткового шару виконується зменшення розмірності за допомогою локального усереднення, створюючи ієрархічну структуру, де останні шари виділяють узагальнені ознаки, менш залежні від спотворень.

Навчання CNN відбувається за допомогою стандартного алгоритму зворотного поширення помилки, але має певні переваги над класичними багат шаровими нейронними мережами. Зокрема, CNN забезпечує значно вищу швидкість і надійність класифікації. Важливою властивістю є здатність використовувати автоматично сформовані ознаки для адаптації до різноманітних варіацій об'єктів, що робить її потужним інструментом для обробки зображень у різних умовах.

На виходах верхніх рівнів ієрархічної архітектури, згорткові нейронні мережі (CNN) демонструють здатність до ефективного витягнення характеристик навіть для зображень, яких не було в навчальному наборі, використовуючи класифікацію за методом найближчих сусідів. Здатність CNN до генералізації дозволяє успішно розпізнавати нові об'єкти завдяки особливій обробці зображень через багатоетапні перетворення.

Нейронна мережа Хопфілда пропонує кардинально інший підхід до навчання в порівнянні з класичними моделями на зразок багат шарових персептронів. Навчання в Хопфілдській мережі відбувається за один цикл, під час якого вагові коефіцієнти обчислюються одночасно за визначеною формулою, що усуває необхідність послідовного налаштування з корекцією похибок. У цій одношаровій архітектурі кожен нейрон з'єднаний з усіма іншими, але зворотні зв'язки між одним і тим самим нейроном відсутні, забезпечуючи своєрідний обмін інформацією в межах мережі.

Відмінною особливістю нейронної мережі Хопфілда є її релаксаційний характер. Після початкової ініціалізації вона автоматично переходить у стан стабільності, що відповідає мінімуму енергії, в процесі поступової релаксації. Під час цього процесу нейрони коригують свої стани, взаємодіючи між собою, поки не досягається рівноважний стан, який залишається незмінним.

Однак, метод має певні обмеження

Візуальні образи, що запам'ятовуються, повинні значно відрізнятися між собою, оскільки надто схожі образи можуть призвести до труднощів у розпізнаванні;

Під час процесу запам'ятовування, зображення не повинні змінювати своє положення, обертатися або зсуватися відносно початкового стану.

Для подолання цих обмежень було запропоновано різні модифікації класичної Хопфілдської мережі. Однією з таких модифікацій є застосування ортогонального перетворення, яке дозволяє адаптувати мережу для роботи з високорельєваними зображеннями. Цей підхід передбачає перетворення початкового набору образів у подвійний набір векторів, що покращує здатність мережі розпізнавати схожі зображення, відновлюючи їх навіть за умов значної кореляції.

Таким чином, нейронна мережа Хопфілда є цікавим і ефективним інструментом для задач пам'яті та відновлення, який демонструє свою унікальну архітектуру та підхід до навчання, що суттєво відрізняється від інших нейромережових моделей.

Прихована марківська модель (ПММ), яку можна уявити як статистичний механізм, слідує динаміці марківського процесу, але з однією суттєвою особливістю: її внутрішні стани залишаються невідомими, прихованими від безпосереднього спостереження, тоді як зовнішні події піддаються реєстрації. У пошуку невідомих параметрів ПММ використовує контрольовані параметри для навчання, що потім застосовуються до задач аналізу й розпізнавання, таких як ідентифікація облич на основі зовнішніх характеристик.

При моделюванні розпізнавання зображень розглядають, що зображення — це двовимірний дискретний сигнал, де вектор спостереження набуває визначальної ролі. Для уніфікації підходів до розпізнавання часто застосовують прямокутні вікна, які накладаються одне на одне, щоб мінімізувати втрату інформації. Оптимальні розміри та ступінь перекриття цих вікон встановлюються експериментально. Після вибору вікна відповідний блок зображення перетворюється на цифровий формат за допомогою методів, таких

як перетворення Карунена-Лоева або дискретне косинусне перетворення (DCT), що дозволяє максимально зберегти характерні ознаки.

Метод опорних векторів (Support Vector Machine, SVM) базується на побудові класифікатора, який не лише розділяє об'єкти по класах, а й максимізує відстань між класами шляхом знаходження оптимальної гіперплощини. Ця гіперплощина вибирається такою, що відстань від неї до найближчих об'єктів (опорних векторів) є максимальною. Це забезпечує ефективність та стійкість класифікації навіть при роботі з даними, що перекриваються. SVM навчається на прикладах, будуючи оптимальну межу між класами, що робить його здатним справлятися з новими, раніше невідомими об'єктами.

Для роботи з нелінійно роздільними класами SVM використовує функції ядра, які проєктують вектори ознак у простір вищої вимірності, роблячи можливим застосування лінійного класифікатора до складних задач. Це дозволяє уникнути великої обчислювальної складності, зберігаючи при цьому високу точність.

Паралельно, метод головних компонент (PCA) використовується для зменшення розмірності простору ознак. Це досягається шляхом перетворення вихідного набору векторів у новий базис, де головна дисперсія даних зосереджена вздовж осей головних компонент, що оптимально описують навчальні дані. Застосування PCA уможливорює зниження вимог до обчислювальних ресурсів та підвищення точності класифікації

Проблеми, що виникають при розпізнаванні облич, можна звести до двох типів помилок: відсутність розпізнавання справжніх облич та помилкове виявлення фальшивих. Оцінка ефективності таких систем включає розрахунок рівня виявлення (процент правильно розпізнаних облич) та рівня хибних спрацювань.

Отже, суть методу опорних векторів полягає в побудові максимально роздільного класифікатора, який знаходить таку гіперплощину, що відстань до

найближчих об'єктів з кожного класу є максимальною, що гарантує найкраще розділення у просторі ознак.

Порівняння точності висновків і величини похибки II роду при використанні описаних методів наведено в таблиці. 1

Таблиця 1.1 - Ефективність методів визначення облич

Метод	Відсоток правильних знаходжень	Помилка другого роду
Нейронні мережі	~92%	~1.3%
Метод опорних векторів	~72%	~0.6%
SNoW	~94%	~0.12%
AdaBoost	~94%	~0.00001%

З огляду на наведені дані, метод AdBoost (бюстінг) виявився кращим з точки зору відсотку правильних виявлень та помилок другого роду. Тому для системи розпізнавання людей у відеопотоках рекомендується використовувати метод Віоли-Джонса, який ґрунтується на цьому підході. Крім високої точності виявлення, цей метод відзначається також високою швидкістю роботи, що робить його найбільш практичним для виявлення осіб у реальному часі.

З огляду на наведені дані, метод AdBoost (бюстінг) виявився кращим з точки зору відсотку правильних виявлень та помилок другого роду. Тому для системи розпізнавання людей у відеопотоках рекомендується використовувати метод Віоли-Джонса, який ґрунтується на цьому підході. Крім високої точності виявлення, цей метод відзначається також високою швидкістю роботи, що робить його найбільш практичним для виявлення осіб у реальному часі.

## 1.2 Огляд та аналіз методів розпізнавання обличчя

Для розпізнавання обличчя існує два основних напрями: методи, що ґрунтуються на значеннях пікселів, та методи, що базуються на виявленні характерних точок [10]. Давайте заглибимося у суть кожної з цих підгруп.

Методи на основі значень пікселів: Ці методи прямо маніпулюють даними пікселів зображення, застосовуючи їх для ідентифікації обличчя. Тут вся інформація вичерпується в числових значеннях, зосереджуючись на матричній структурі зображення.

Методи на основі характерних точок: Ці методи зосереджені на виявленні ключових точок та унікальних особливостей обличчя. Вони аналізують геометричні відстані і взаємозв'язки між рисами, такими як очі, ніс та рот, щоб створити більш абстрактне представлення.

Ці дві підгрупи пропонують контрастні, але взаємодоповнюючі підходи до розпізнавання облич, використовуючи як детальну піксельну інформацію, так і специфічні риси обличчя. Це дозволяє ефективно виявляти та ідентифікувати обличчя в зображеннях, відкриваючи нові горизонти для технологій розпізнавання.

### **1.2.1 Методи на основі значень яскравості пікселів**

Ця група методів, що використовує значення пікселів, відома під назвою методи на основі значень пікселів. Її назва говорить сама за себе: для розпізнавання обличчя застосовуються дані про колір та яскравість пікселів зображення. Найбільш елементарним методом у цій категорії є просте порівняння, де подібність оцінюється через відстань між векторами яскравості пікселів. Проте, цей підхід виявляється чутливим до змін освітлення, позиції обличчя та масштабу. Крім того, його обчислювальна складність є високою, що робить його непридатним для застосувань у реальному часі. Тому для більш ефективного порівняння часто вдаються до методів, що зменшують розмірність

векторних описів зображень, що суттєво знижує обчислювальну навантаженість і підвищує швидкість розпізнавання [10].

Одним з найпомітніших методів є Eigenfaces, запропонований Метью Турком і Алексом Пентландом у 1991 році [11]. Цей метод здобув популярність як перший ефективний засіб для розпізнавання облич. Основна концепція полягає в застосуванні методу головних компонент для виявлення векторів, які найкраще відображають обличчя людей. Використовуючи цей підхід, можна визначити різноманітні зміни в наборі навчальних зображень облич і описати їх у базисі ортогональних векторів, відомих як власні обличчя. Ці власні обличчя мають приголомшливу властивість: зображення, що відповідає кожному вектору, має форму обличчя.

Обчислення головних компонент полягає у визначенні власних векторів і власних значень коваріаційної матриці, розрахованої на основі зображень. Для кодування інших зображень обличчя використовується набір власних векторів, отриманих на навчальному наборі, для представлення нових зображень через зважену комбінацію цих векторів. З обмеженою кількістю власних векторів можна отримати компактне представлення вхідного зображення обличчя, яке зберігається в базі даних у вигляді вектора коефіцієнтів, слугуючи ключем для пошуку в базі даних облич [11]. Комбінація головних компонент та відповідних власних векторів забезпечує реконструкцію зображення обличчя.

Для кожного зображення обличчя обчислюються його основні компоненти, які зазвичай коливаються від 5 до 200. Процес розпізнавання ґрунтується на порівнянні основних компонентів невідомого зображення з компонентами відомих зображень. При цьому вважається, що зображення облич, що належать одній людині, формують кластери в просторі основних компонентів. З бази даних обираються кандидати-зображення, що мають найменшу відстань до вхідного зображення.

Проте використання методу власних облич вимагає дотримання певних ідеалізованих умов, таких як рівномірне освітлення, нейтральний вираз обличчя

та відсутність перешкод, таких як окуляри чи борода. Якщо ці умови не дотримуються, основні компоненти можуть не відобразити міжкласові варіації. Наприклад, в умовах змінного освітлення метод власних облич може виявитися малоефективним, адже перші головні компоненти часто відображають зміни освітлення, що призводить до неточних результатів.

Однак за дотримання ідеальних умов точність розпізнавання може перевищувати 90%. Обчислення власних векторів є досить трудомістким процесом. Один зі способів їх отримання полягає в розгортанні зображення у вигляді стрічок, що дозволяє зменшити його розмір, прискорюючи обчислення та розпізнавання, хоча й не дозволяючи точно відновити початкове зображення.

Метод головних компонент виявляється ефективним у практичних застосуваннях. Однак у ситуаціях, коли зображення обличчя підлягає значним змінам освітлення або виразу, його ефективність суттєво знижується. Основна ідея методу полягає в тому, що зображення обличчя розкладається на набір векторів основних компонентів, які найкраще описують варіації даних. Кожен вектор основної компоненти представляє певну характеристику або шаблон, який може бути використаний для подальшого порівняння та розпізнавання облич.

#### Переваги:

- Простий у реалізації, що дозволяє легко впроваджувати його у різних застосуваннях;
- Ефективний у розпізнаванні облич у реальному часі;
- Вектори компонентів можуть бути компактно збережені
- Розпізнавання виконується швидко.

#### Недоліки:

- Висока трудомісткість розрахунку власних векторів.
- Вимога до відсутності перешкод, таких як окуляри або борода.
- Вимога до умов близького освітлення та однакової перспективи.
- Необхідність повторного навчання при додаванні нових облич у базу.

Недоліки класичного методу Eigenfaces мають свої корені в тому, що головні власні вектори, що визначають підпростір облич, частіше виділяють деталі освітлення, аніж індивідуальні риси облич. Це пов'язано з тим, що метод головних компонентів (PCA) орієнтований на створення підпростору для наближення загальних даних, а не на розпізнавання конкретних класів, тому характерні особливості облич можуть "розчинитися" у таких векторах.

Метод Fisherfaces, що базується на лінійному дискримінантному аналізі (LDA), дозволяє ефективніше розділяти класи зображень, оскільки він спрямований на пошук проекцій, що максимально відділяють один клас від іншого. Це дає змогу виявити проекції, які розташовують образи одного класу ближче одне до одного, а образи різних класів максимально розділяють. Завдяки цьому метод демонструє високу стійкість до змін освітлення — але з умовою, що освітлення під час навчання моделі та її використання буде схожим.

Переваги Fisherfaces:

- Розпізнавання облич за умов різного освітлення: метод дозволяє точно визначати обличчя, незважаючи на зміну рівнів освітлення завдяки обробці даних з різних перспектив.
- За рахунок багаточисельних зображень кожної особи у вибірці, Fisherfaces може значно знижувати вплив зовнішніх факторів.

Недоліки Fisherfaces:

- Відсутність інваріантності щодо афінних перетворень: цей метод не може коректно працювати з деформаціями, що виникають, наприклад, через нахил обличчя.

Метод локальних бінарних шаблонів (LBP), створений для класифікації текстур у 1996 році, отримав популярність і в задачах розпізнавання облич. Його простота та ефективність дозволяють швидко перетворювати зображення, аналізуючи яскравість сусідніх пікселів і присвоюючи кожному значення, що відображає локальні структури текстур. Результуюче зображення поділяється на

регіони, для кожного з яких формується гістограма, а потім вони об'єднуються для остаточного порівняння з іншими зображеннями в базі даних за допомогою методів машинного навчання.

Переваги LBP:

- Простота реалізації та швидкодія: метод легко інтегрується в системи реального часу, не потребуючи значних обчислювальних ресурсів.
- Висока стійкість до монотонних змін освітлення, що забезпечує стабільність у задачах ідентифікації облич навіть за змінних світлових умов.
- Ці властивості роблять метод LBP привабливим для використання у відеоспостереженні та інших системах реального часу, де важливі швидкість і точність обробки обличчя.

### **1.2.2 Методи на основі характерних точок**

У класі методів, що покладаються на характерні точки, обробка зображень переходить від аналізу яскравості пікселів до аналізу просторового розташування ключових точок, таких як очі, лінії брів, ніс, рот та інші особливості обличчя. Це включає моделі активного вигляду та активні моделі форм, здатні надійніше ідентифікувати та відслідковувати обличчя навіть за наявності деталей, як-от вуса чи окуляри.

Активні моделі зовнішнього вигляду (AAM), запроваджені Тімом Кутсом і Крісом Тейлором у 1998 році, використовують деформаційно-статистичний підхід, щоб адаптувати зображення обличчя до реалістичних умов. Ці моделі застосовують параметри форми для визначення контурів об'єкта та параметри вигляду для збереження статистичних ознак текстури та яскравості. Адаптація моделі до реального зображення починається з розмітки контрольних точок, виконуваної вручну, щоб максимально точно навчити систему на вибірці. На початковому етапі використовують маску для локалізації обличчя, що дозволяє

згодом визначити його основні точки, а потім відслідковувати розмір і положення обличчя на зображенні, витягуючи необхідні риси з моделі. Завдяки цьому підходу обличчя може бути ідентифіковано, навіть якщо присутні певні перешкоди чи незначні видозміни.

Активні моделі форм (ASM) також базуються на статистичному аналізі, але більше зосереджені на формі об'єкта. Цей метод передбачає нанесення антропометричних точок на зображення, що уможливорює деталізований аналіз обличчя з урахуванням його індивідуальних особливостей. За допомогою прокрустового аналізу точки масштабується, центруються та обчислюється середнє значення форми й коваріаційна матриця для всього набору об'єктів. Далі, розташування антропометричних точок нового обличчя визначається шляхом вирішення оптимізаційної задачі. Цей підхід забезпечує точну локалізацію ключових характеристик, яка дозволяє підвищити точність розпізнавання в подальших етапах обробки.

Примітно, що ASM орієнтовані не на саме розпізнавання, а на локалізацію точок, що дозволяє стандартизувати розташування та масштаб для подальшого аналізу. Зазвичай, для локалізації використовують обмежену кількість точок, що сприяє підвищенню швидкості обробки, проте для більш точної ідентифікації потрібно додатково включати більшу кількість контрольних точок, що може знижувати продуктивність системи .

На основі аналізу можна стверджувати, що метод локальних бінарних шаблонів (LBP) демонструє найкраще співвідношення швидкості обробки та точності, що робить його оптимальним для застосувань у реальному часі. У наступних розділах буде докладніше розглянуто цей метод і обговорено ефективність різних його модифікацій для визначення оптимального підходу у розпізнаванні облич, враховуючи високу точність та швидкість роботи системи.

### **1.3 Що таке нейронні мережі**

Нейронні мережі є фундаментальними будівельними блоками алгоритмів глибокого навчання. Нейронна мережа — це тип алгоритму машинного навчання, розроблений для моделювання поведінки людського мозку. Він складається з взаємопов'язаних вузлів, також відомих як штучні нейрони, які організовані в шари. (рис. 1.1) Таким чином можна отримати адаптивну систему, за допомогою якої комп'ютери навчаються на своїх помилках та постійно вдосконалюються. Штучні нейронні мережі намагаються вирішувати складні завдання, такі як резюмування документів або розпізнавання осіб з більш високою точністю.

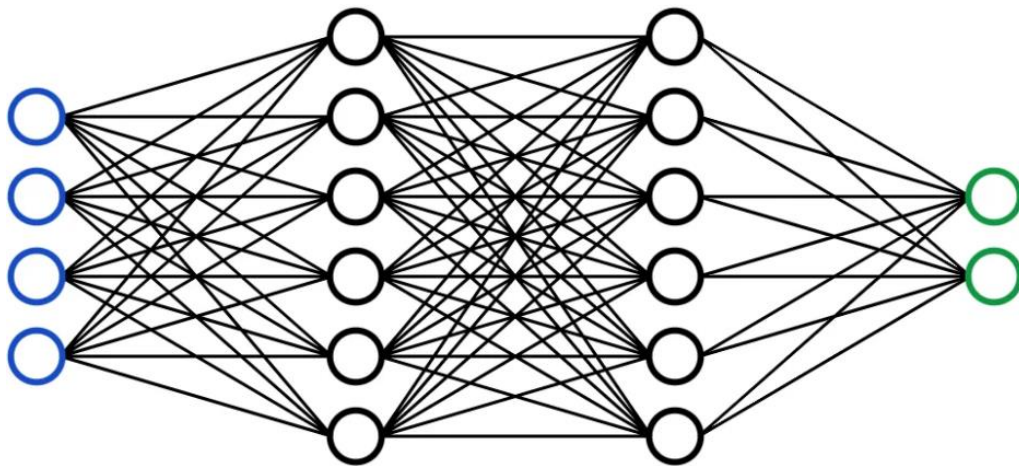


Рисунок 1.1 – Схематичне зображення шарів нейронів

Нейронні мережі — це засіб машинного навчання, за якого комп'ютер вчиться виконувати певне завдання, аналізуючи навчальні приклади. Зазвичай приклади заздалегідь позначаються вручну. Система розпізнавання об'єктів, наприклад, може отримати тисячі позначених зображень автомобілів, будинків, кавових чашок тощо, і вона знайде візуальні моделі в зображеннях, які послідовно корелюють з певними мітками.

За моделлю людського мозку нейронна мережа складається з тисяч або навіть мільйонів простих вузлів обробки, які тісно пов'язані між собою. Більшість сучасних нейронних мереж організовано в шари вузлів, і вони є «прямими», тобто дані переміщуються через них лише в одному напрямку. Окремий вузол може бути з'єднаний з декількома вузлами на рівні під ним, з

якого він отримує дані, і з кількома вузлами на рівні над ним, до якого він надсилає дані.

Кожному зі своїх вхідних з'єднань вузол призначає номер, відомий як «вага». Коли мережа активна, вузол отримує інший елемент даних — інше число — через кожне зі своїх з'єднань і множить його на відповідну вагу. Потім він додає отримані продукти разом, отримуючи єдине число. Якщо це число нижче порогового значення, вузол не передає дані на наступний рівень. Якщо число перевищує порогове значення, вузол «спрацьовує», що в сучасних нейронних мережах зазвичай означає надсилання числа — суми зважених вхідних даних — по всіх вихідних з'єднаннях.

Коли нейронна мережа навчається, усі її ваги та пороги спочатку встановлюються на випадкові значення. Навчальні дані подаються на нижній рівень — вхідний рівень — і проходять через наступні шари, множаться та додаються складними способами, доки, нарешті, радикально перетворені не надходять на вихідний рівень. Під час навчання вагові коефіцієнти та порогові значення постійно коригуються, доки навчальні дані з однаковими мітками не отримають однакові результати.

### **1.3.1 Сфери застосування нейронних мереж**

У наш час нейронні мережі дуже розповсюджені й використовуються в багатьох сферах нашого життя. Важко уявити людину, яка не користувалася чи хоча б не чула про штучний інтелект. Головні напрямки застосування нейромереж:

- Діагностика за допомогою класифікації медичних зображень
- Цільовий маркетинг за допомогою фільтрації соціальних мереж та аналізу поведінкових даних

- Фінансові прогнози за допомогою обробки історичних даних фінансових інструментів
- Прогнозування електричного навантаження та потреби в енергії
- Контроль відповідності вимогам та якості
- Визначення хімічних сполук

Також окремо необхідно відмітити чотири сфери в яких вплив нейронних мереж має найвищий вплив і актуальність.

Машинний зір – це здатність комп'ютерів отримувати інформацію та сенс із зображень та відео. За допомогою нейронних мереж комп'ютери можуть розрізняти та розпізнавати зображення так, як це роблять люди. Машинний зір застосовується в кількох областях, наприклад візуальне розпізнавання у безпілотних автомобілях, щоб вони могли реагувати на дорожні знаки та інших учасників руху. Модерація контенту для автоматичного видалення небезпечного або неприйняттого контенту з архівів зображень та відео. Розпізнавання осіб для ідентифікації людей та розпізнавання таких атрибутів, як відкриті очі, окуляри та рослинність на обличчі. Маркування зображення для ідентифікації логотипів бренду, одягу, захисного спорядження та інших деталей зображення.

Розпізнавання мови. Нейронні мережі можуть аналізувати людську мову незалежно від її мовних моделей, висоти, тону, мови та акценту. Віртуальні помічники, такі як Amazon Alexa та програмне забезпечення для автоматичної транскрипції, використовують розпізнавання мови для виконання наступних завдань: допомога операторам кол-центру та автоматична класифікація дзвінків, перетворення клінічних рекомендацій на документацію в режимі реального часу, точні субтитри до відео та записів нарад для ширшого охоплення контенту

Обробка природної мови (NLP) - це здатність обробляти природний, створений людиною текст. Нейронні мережі допомагають комп'ютерам отримувати інформацію та зміст із текстових даних та документів. NLP має кілька сфер застосування, у тому числі:

- Автоматизовані віртуальні агенти та чат-боти.
- Автоматична організація та класифікація записаних даних

- Бізнес-аналітика довгих документів: наприклад, електронних листів та форм
- Індексція ключових фраз, що вказують на настрої: наприклад, позитивних та негативних коментарів у соціальних мережах
- Узагальнення документів та генерація статей з заданої теми

Сервіси рекомендацій. Нейронні мережі можуть відстежувати дії користувачів для розробки персональних рекомендацій. Вони також можуть аналізувати всі дії користувачів та виявляти нові продукти чи послуги, які цікавлять конкретного споживача. Наприклад, стартап із Філадельфії Curalate допомагає брендам конвертувати повідомлення у соціальних мережах у продажі. Бренди використовують службу інтелектуального маркування продуктів (IPT) Curalate для автоматизації збору та обробки контенту користувачів соціальних мереж. IPT використовує нейронні мережі для автоматичного пошуку та рекомендації продуктів, що відповідають активності користувача у соціальних мережах. Споживачам не потрібно копатися в онлайн-каталогах, щоб знайти конкретний продукт із зображення в соціальних мережах. Натомість вони можуть використовувати автоматичне маркування Curalate, щоб з легкістю придбати продукт.

### **1.3.2 Принцип роботи нейронних мереж**

Архітектура нейронних мереж повторює структуру людського мозку. Клітини людського мозку, звані нейронами, утворюють складну мережу з високим ступенем взаємозв'язку та посиляють один одному електричні сигнали, допомагаючи людям обробляти інформацію. Так само штучна нейронна мережа складається з штучних нейронів, які взаємодіють на вирішення проблем. Штучні нейрони - це програмні модулі, які називаються вузлами, а штучні нейронні мережі - це програми або алгоритми, які використовують обчислювальні системи для виконання математичних обчислень.

Кожен нейрон представляє одиницю обчислень, яка приймає набір вхідних даних, виконує набір обчислень і створює вихідні дані, які передаються на наступний рівень.

Як і нейрони в нашому мозку, кожен вузол нейронної мережі отримує вхідні дані, обробляє їх і передає вихідні дані наступному вузлу. Коли дані переміщуються мережею, зв'язки між вузлами зміцнюються або послаблюються залежно від шаблонів у даних. Це дозволяє мережі вчитися на даних і робити прогнози або рішення на основі отриманих даних.

Уявіть сітку 28 на 28, де число намальовано таким чином, що одні пікселі темніші за інші. Визначивши яскравіші пікселі, ми можемо розшифрувати число, написане на сітці. Ця сітка служить входом для нейронної мережі. (рис. 1.2)

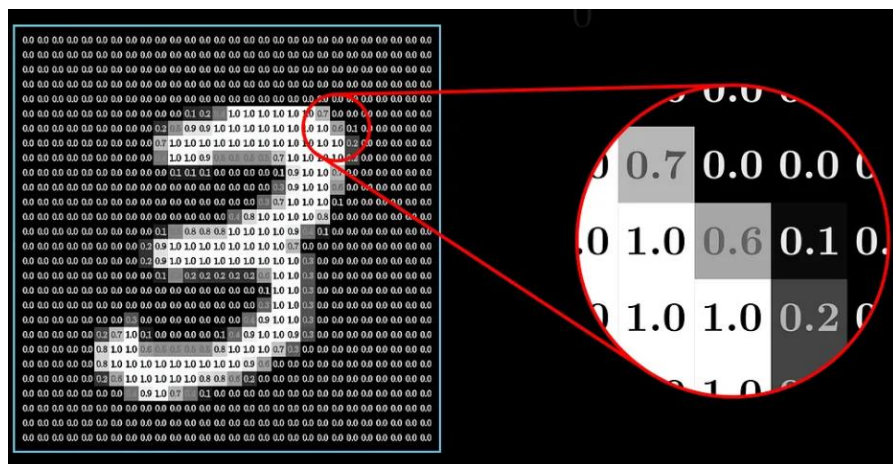


Рисунок 1.2 – Розшифрування числа за яскравістю пікселів

Рядки сітки розташовані в горизонтальному 1-D масиві, який потім перетворюється на вертикальний масив, утворюючи перший шар нейронів

У випадку першого шару кожен нейрон відповідає одному пікселю у вхідному зображенні, а значення всередині кожного нейрона представляє активацію або інтенсивність цього пікселя. Вхідний рівень нейронної мережі відповідає за прийом необроблених даних (у даному випадку зображення) і перетворення їх у формат, який може оброблятися рештою мережі.

У цьому випадку ми маємо вхідні пікселі 28x28, що дає нам загалом 784 нейрони у вхідному шарі. (рис. 1.3) Кожен нейрон матиме значення активації 0 або 1 залежно від того, чорним чи білим є відповідний піксель у вхідному зображенні відповідно.

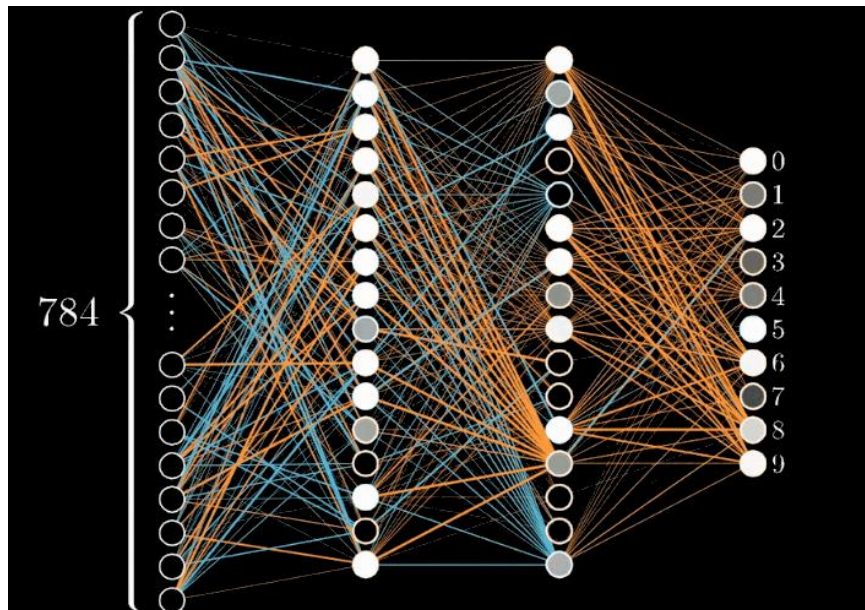


Рисунок 1.3 – Нейронна сітка

Вихідний рівень нейронної мережі в цьому випадку складається з 10 нейронів, кожен з яких представляє можливий клас виходу (у цьому випадку цифри від 0 до 9). Вихід кожного нейрона у вихідному шарі представляє ймовірність того, що вхідне зображення належить до цього конкретного класу. Найвище значення ймовірності визначає прогнозований клас для цього вхідного зображення.

Приховані шари. (рис. 1.4) Між вхідним і вихідним шарами ми маємо один або кілька прихованих шарів, які виконують ряд нелінійних перетворень вхідних даних. Метою цих прихованих шарів є вилучення функцій вищого рівня з вхідних даних, які є більш значущими для поточного завдання. Ви самі вирішуєте, скільки прихованих шарів ви хочете додати у свою мережу.

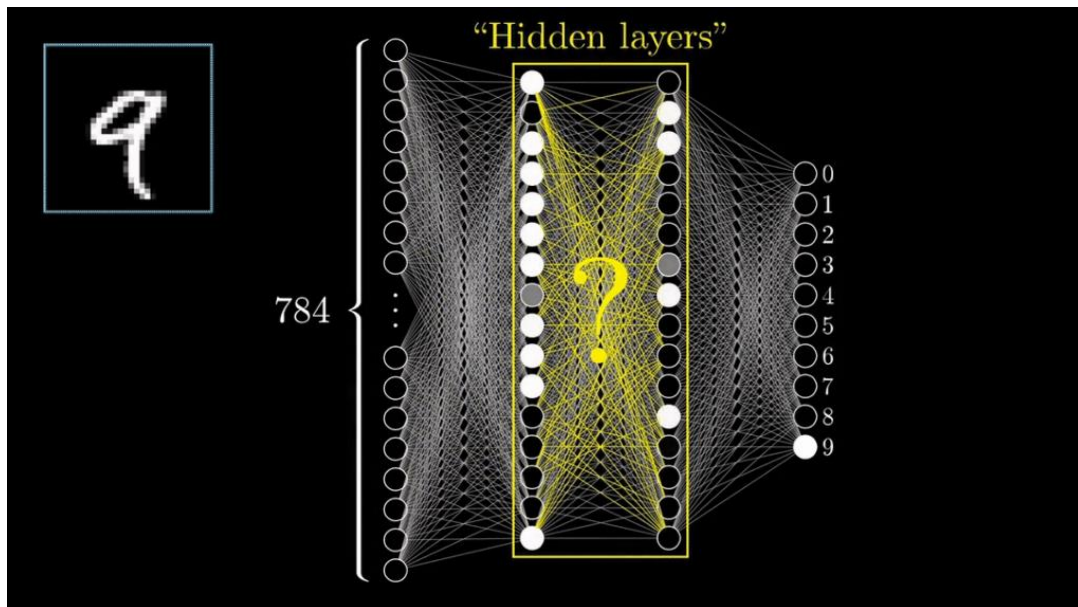


Рисунок 1.4 – Нейронна сітка

Кожен нейрон прихованого шару отримує вхідні дані від усіх нейронів попереднього шару та застосовує до цих вхідних даних набір вагових коефіцієнтів і зміщень перед тим, як передати результат через нелінійну функцію активації. Цей процес повторюється для всіх нейронів прихованого шару, доки не буде досягнуто вихідного шару.

Пряме поширення. Пряме розповсюдження – це процес, за допомогою якого вхідні дані передаються через нейронну мережу для генерування виходу. Він передбачає обчислення виходу кожного нейрона на кожному рівні мережі шляхом застосування вагових коефіцієнтів і зміщень до вхідних даних і передачі результатів через функцію активації.

$$y = g(w_0 + \text{Sigma } x_i \cdot w_i) \text{ or } y = g(w_0 + X \cdot W)$$

де  $y$  — вихід нейронної мережі,  $g$  — нелінійна функція активації,  $x_i$  —  $i$ -та вхідна функція або вхідна змінна,  $w_i$  — вага, пов'язана з  $i$ -тою вхідною ознакою або змінною,  $w_0$  — член зсуву, який є постійним значенням, яке додається до лінійної комбінації вхідних даних.

$\Sigma w_i x_i$ .  $w_i$ : це лінійна комбінація вхідних ознак і пов'язаних з ними ваг. Цей термін також іноді називають «зваженою сумою» вхідних даних.

Зворотне поширення — популярний алгоритм, який використовується для навчання нейронних мереж. Він передбачає обчислення градієнта, який є мірою зміни функції втрат щодо кожної ваги в мережі. Функція втрат є показником того, наскільки добре нейронна мережа здатна передбачити правильний вихід для заданого вхідного сигналу. Обчислюючи градієнт функції втрат, зворотне поширення дозволяє нейронній мережі оновлювати свої ваги таким чином, щоб зменшити загальну помилку або втрату під час навчання.

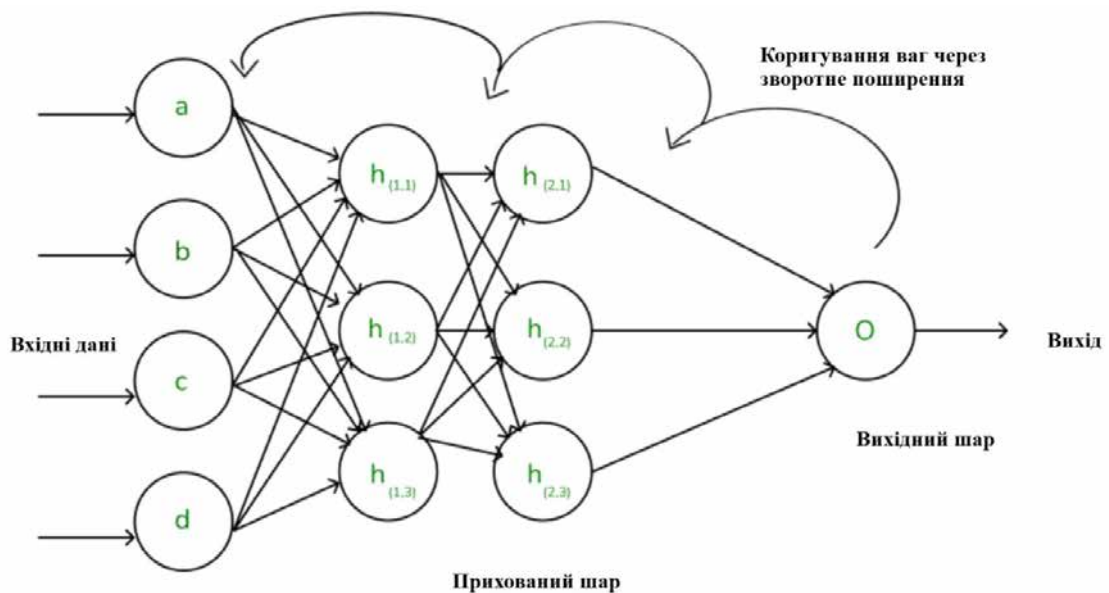


Рисунок 1.5 – Зворотне поширення

Алгоритм працює шляхом поширення помилки від вихідного рівня назад через рівні мережі, використовуючи ланцюгове правило обчислення для обчислення градієнта функції втрат щодо кожної ваги. Потім цей градієнт використовується в оптимізації градієнтного спуску для оновлення вагових коефіцієнтів і мінімізації функції втрат.

### 1.3.3 Термінології, що використовуються в нейронних мережах

Навчання нейронної мережі — це процес коригування вагових коефіцієнтів нейронної мережі на основі вхідних даних і бажаного результату з метою підвищення точності передбачень мережі.

Вага: ваги стосуються параметрів, які вивчаються під час навчання, і вони визначають міцність зв'язків між нейронами. Кожному з'єднанню між нейронами присвоюється вага, який множиться на вхідне значення нейрона, щоб визначити його вихід.

Зміщення: зміщення – це ще один вивчений параметр, який додається до зваженої суми вхідних даних для нейрона на даному шарі. Це додатковий вхід для нейрона, який допомагає регулювати вихід функції активації. Зміщення може стосуватися будь-яких фіксованих факторів, які впливають на прибуток продукту, але не пов'язані безпосередньо з ціною чи маркетинговими витратами.

Нелінійна функція активації: нелінійна функція активації застосовується до вихідного сигналу нейрона, щоб ввести нелінійність у мережу. Нелінійність важлива, оскільки вона дозволяє мережі моделювати складні, нелінійні зв'язки між входами та виходами. Загальні функції активації, які використовуються в нейронних мережах, включають сигмоїдну функцію, функцію ReLU (Rectified Linear Unit) і функцію softmax.

Функція втрат: це математична функція, яка вимірює помилку або різницю між прогнозованим виходом нейронної мережі та справжнім виходом. Емпіричні втрати вимірюють загальні втрати для всього нашого набору даних. Перехресна втрата ентропії зазвичай використовується з моделями, які виводять ймовірність від 0 до 1, тоді як середня квадратична втрата помилки використовується з регресійними моделями, які виводять безперервні дійсні числа. Мета полягає в тому, щоб мінімізувати функцію втрат під час навчання, щоб підвищити точність прогнозів мережі. Наприклад, якщо товар є сезонним товаром, у певну пору року може існувати тенденція до більшого прибутку. Різниця між фактичним прибутком і прогнозованим прибутком є функцією збитків

Оптимізація втрат: це процес мінімізації помилок або втрат, понесених нейронною мережею під час прогнозування. Це робиться шляхом регулювання ваг мережі.

Градiєнтний спуск: це алгоритм оптимізації, який використовується для знаходження мінімуму функції, наприклад функції втрат нейронної мережі. Він передбачає ітераційне коригування вагових коефіцієнтів у напрямку від'ємного градієнта функції втрат. Ідея полягає в тому, щоб продовжувати рухати ваги в напрямку, який зменшує втрати, доки ми не досягнемо мінімуму. Градієнтний спад передбачає обчислення градієнта функції втрат щодо кожної вхідної функції та використання цього градієнта для ітеративного коригування значень ознак, доки не будуть знайдені оптимальні значення, а також процес, який передбачає пошук оптимальних значень для вхідних функцій для мінімізації втрат.

#### **1.4 Постановка мети і завдань дослідження**

Розроблена система для обробки відеопотоку базується на багаторівневому підході, де відбувається поетапна обробка інформації. Перший рівень – це швидка ідентифікація облич, реалізована за допомогою алгоритму Віюлі-Джонса. Другий – більш деталізоване розпізнавання виявлених облич, здійснюване на основі моделі LBPН і натренованої нейронної мережі. Залежно від рівнів освітлення та ракурсів зображень, точність системи може коливатися, тому потрібні суворіші параметри для коректної роботи:

- Однорідність освітлення: система адаптована до випадків із монотонними змінами освітлення, а отже, усі навчальні й тестові зразки будуть збиратися в подібних умовах.
- Фронтальна перспектива обличчя: система оптимізована для роботи із зображеннями, де обличчя розташовані фронтально або з незначними відхиленнями й без яскравих емоційних виразів.

- Відсутність сторонніх об'єктів: система передбачає, що обличчя на кадрах будуть без перекриттів або мінімально перекриті.
- Робота в реальному часі: обробка здійснюється в режимі реального часу завдяки підключеній до комп'ютера камері, що забезпечує динамічність процесу розпізнавання.
- Вивід результатів: система виводить інформацію про виявлене обличчя, зокрема ім'я людини з бази даних.

Для підвищення якості розпізнавання, доцільно застосовувати кілька додаткових етапів обробки. Зокрема, після первинного виявлення облич, обробка фільтром Гауса здатна зменшити вплив шумів і підвищити чіткість зображення, що сприяє точності подальшого розпізнавання.

## **1.5 Висновки з розділу 1**

У першому розділі дослідження окреслено цільове призначення та завдання, що впливають із докладного аналізу сучасних методів ідентифікації облич, а також проведено ретельне порівняння їхніх сильних і слабких сторін.

Надано детальне пояснення терміну нейронні мережі. Підкреслено важливість і актуальність застосування нейронних мереж у різних сферах життя. Розкрито тему структури нейронної мережі та принципи її функціонування на прикладах.

У завершенні розділу сформульовано мету та завдання дослідження. Створювана система базуватиметься на багаторівневому підході. Проаналізовано можливі слабкі сторони та вимоги до майбутньої системи.

## **2 ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ**

### **2.1 Виявлення обличчя методом Віюлі-Джонса**

Метод Віоли-Джонса, створений ще в 2001 році [3], зберіг своє центральне місце серед алгоритмів обробки зображень у реальному часі, і за цей час став основою для різноманітних завдань, від розпізнавання облич до більш комплексних систем відеоспостереження та пошуку об'єктів. Незважаючи на вік технології, її популярність лише підтверджує доцільність і надійність підходу, який залишається актуальним навіть сьогодні.

Основні принципи, на яких ґрунтується метод Віоли-Джонса, можна подати через декілька ключових етапів:

- Представлення зображення у цілісній формі для подальшої обробки;
- Застосування особливостей Хаара для пошуку облич;
- Використання каскадної класифікації та методу бустингу.

### 2.1.1 Розпізнавання обличчя методом Віоли-Джонса

Важливою складовою методу є інтегральне представлення зображення, яке істотно пришвидшує обчислення яскравостей прямокутних областей. Завдяки інтегральній матриці, розмір якої відповідає вихідному зображенню, можна швидко знаходити суми інтенсивностей у різних прямокутних ділянках. Кожен елемент цієї матриці вказує на суму значень пікселів, які розташовані ліворуч і вище від обраного елемента, що дає змогу миттєво обчислювати певні ознаки незалежно від їхніх геометричних параметрів.

З математичної точки зору, кожен елемент інтегрального зображення

$I(x,y)$  розраховується за формулою, де значення інтенсивності пікселя визначає суму інтенсивностей зони вище та лівіше поточної точки:

$$I(x,y) = \sum_{X < x',y < y'} i(x',y'), \quad (1)$$

$$X < x,y < y$$

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

Використання інтегрального представлення зображення дозволяє одночасно обчислювати ознаки одного типу, але різні геометричні параметри. Це можливо завдяки тому, що розрахунок інтегральної матриці займає лінійний час, що пропорційний кількості пікселів у зображенні.

### 2.1.2 Ознаки Хаара

Ознаки Хаара — це особливості цифрового зображення, що стали основою для методів розпізнавання образів, зокрема в задачах ідентифікації облич. Їхня назва відсилає до відомих вейвлетів Хаара, з якими вони мають певну візуальну і математичну схожість. Саме ці ознаки стали основою першого реального часу детектора облич[19].

Концепція використання вейвлетоподібних ознак для розпізнавання об'єктів була вперше висунута Папагорджу в 1998 році [20]. Пізніше цю ідею розвинули Джонс та Віола. Вони модифікували її, увівши у свою систему прямокутні особливості, що нині відомі як ознаки Хаара[3]. Ці ознаки можна побачити на рисунку 2.1, вони використовуються для зосередження уваги на певних регіонах зображення, забезпечуючи високоефективне розпізнавання, адаптоване до задач комп'ютерного зору у режимі реального часу.

В оригінальній версії алгоритму Віоли-Джонса використовувалися лише примітиви без поворотів, а для обчислення значення ознаки сума яскравостей пікселів однієї підобласті віднімали від суми яскравостей іншої підобласті. У розвитку способу були запропоновані примітиви з нахилом на 45 градусів і несиметричних змін. Також замість обчислення звичайної різниці, було запропоновано приписувати кожній підобласті певну вагу та значення ознаки обчислювати як зважену суму пікселів різнотипних областей.

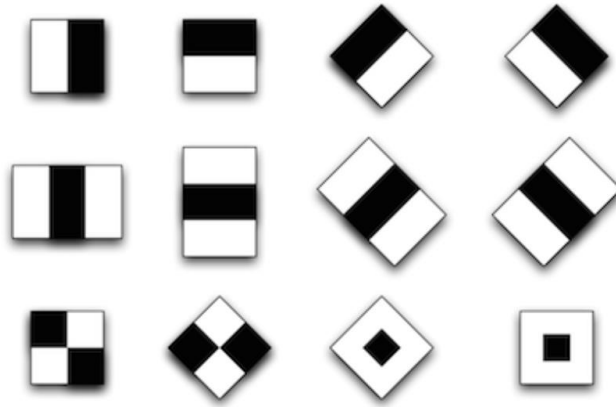


Рисунок 2.1 - Ознаки Хаара

У розширеному методі Віоли-Джонса, який використовується в бібліотеці комп'ютерного зору OpenCV, а також в нашій розробленій системі, доступні додаткові можливості. Результатом обчислення такої ознаки на інтегральному представленні буде:

$$F = U - V,$$

де  $U$  - сума значень яскравості точок, покритих світлою частиною знака, а  $V$  - сума значень яскравості точок, покритих темною частиною знака. Такі особливості описують різницю в яскравості вздовж двох осей зображення.

Для виявлення обличч у зображенні застосовується скануюче вікно з розміром 24x24 пікселі. Цей невеликий фрейм ковзає по всьому зображенню з кроком у 1 піксель. У кожній позиції цього вікна виконуються обчислення ознак Хаара, що можуть мати різні масштаби та конфігурації у межах скануючого вікна. Процес виявлення почерогово повторюється для різних масштабів самого вікна, дозволяючи виявляти обличчя людини незалежно від їх розмірів у кадрі.

Скановані ознаки передаються класифікатору, який вирішує, чи містить аналізована область обличчя або ж вона є неактуальною. Цей метод дозволяє детально проаналізувати зображення та забезпечує високу точність у розпізнаванні, незалежно від масштабів або положення об'єктів.

### 2.1.3 Каскадна класифікація

Каскадна структура класифікатора значно прискорює процес виявлення облич, концентруючись на найбільш перспективних областях зображення. Це досягається за допомогою послідовності слабких класифікаторів, що структуруються каскадом і проходять навчання через алгоритм посилення (boosting). Каскад дозволяє швидко відсівати області без облич ще на початкових етапах, що зменшує обчислювальні навантаження. На рисунку 2.2 наведено приклад каскадної структури класифікаторів.

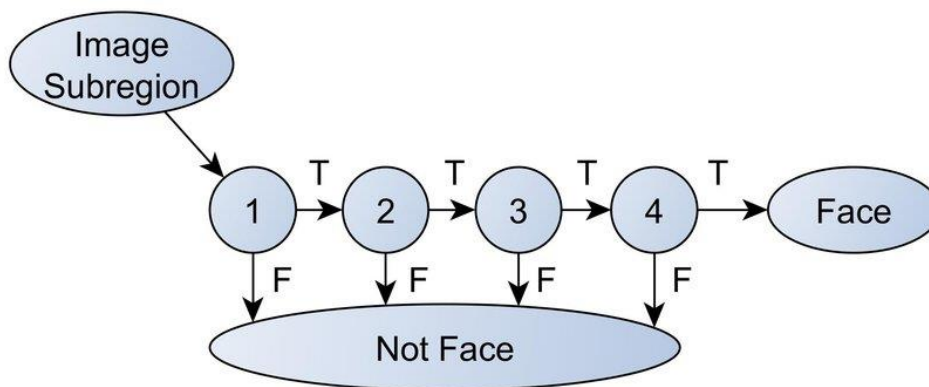


Рисунок 2.2 – Структура каскадного класифікатора

Навчання на кожному рівні каскаду виконується за алгоритмом AdBoost, який підсилює ефективність класифікатора. Під час навчання до кожного рівня додаються нові ознаки, доки не досягнеться оптимальний рівень точності виявлення та мінімізації помилок першого роду. Рівні каскаду визначаються шляхом перевірки на тестовому наборі: якщо задана похибка для об'єкта ще не досягнута, до каскаду додають новий шар. Негативний набір для навчання наступного шару створюється шляхом збору усіх помилкових спрацьовувань поточного рівня.

Результатом роботи каскаду є множина областей зображення, що можуть містити обличчя. Щоб уникнути дублювання однієї й тієї ж області через

масштабування скануючого вікна, повторні виявлення виключаються. Після виявлення обличчя підлягають подальшій обробці — перетворенню у градації сірого і масштабуванню до розмірів 128x128 пікселів для подальшого аналізу.

Каскадна структура класифікатора значно прискорює процес виявлення облич, концентруючись на найбільш перспективних областях зображення. Це досягається за допомогою послідовності слабких класифікаторів, що структуруються каскадом і проходять навчання через алгоритм посилення (boosting). Каскад дозволяє швидко відсівати області без облич ще на початкових етапах, що зменшує обчислювальні навантаження.

Навчання на кожному рівні каскаду виконується за алгоритмом AdBoost, який підсилює ефективність класифікатора. Під час навчання до кожного рівня додаються нові ознаки, доки не досягнеться оптимальний рівень точності виявлення та мінімізації помилок першого роду. Рівні каскаду визначаються шляхом перевірки на тестовому наборі: якщо задана похибка для об'єкта ще не досягнута, до каскаду додають новий шар. Негативний набір для навчання наступного шару створюється шляхом збору усіх помилкових спрацьовувань поточного рівня.

Результатом роботи каскаду є множина областей зображення, що можуть містити обличчя. Щоб уникнути дублювання однієї й тієї ж області через масштабування скануючого вікна, повторні виявлення виключаються. Після виявлення обличчя підлягають подальшій обробці — перетворенню у градації сірого і масштабуванню до розмірів 128x128 пікселів для подальшого аналізу.

## **2.2 Фільтр Гауса**

Фільтр Гауса слугує інструментом для згладжування шумів у зображеннях облич, використовуючи метод м'якого розмиття, що ґрунтується на нормальному розподілі. Цей фільтр застосовує спеціальне перетворення, яке вираховується

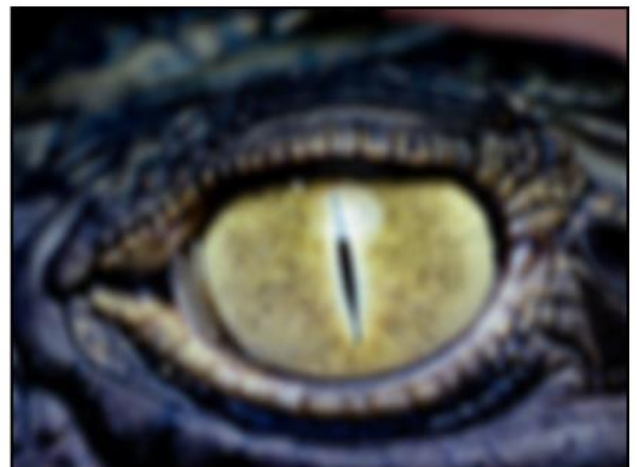
для кожного пікселя зображення, щоб створити ефект розмиття. У двовимірному випадку формула нормального розподілу визначає поверхню, що складається з концентричних кіл, в яких інтенсивність спадає від центральної точки. Нормальний розподіл для двох вимірів описується формулою [21]:

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

Формула цього розподілу, яка лежить в основі згортки, створює матрицю, що накладається на вхідне зображення. В результаті, центральний піксель отримує максимальну вагу (найвище значення функції Гауса), тоді як сусідні пікселі отримують знижені ваги залежно від своєї віддаленості від центру. Цей процес дозволяє забезпечити точніше згладжування і зниження випадкових шумів, зберігаючи при цьому контури обличчя. На рисунку 2.3 наведено приклад роботи одноосного фільтра Гауса, який демонструє ефект поступового розмиття.



Оригінальне зображення



Застосування фільтра Гауса

Рисунок 2.3 – Гаусове розмиття на одновимірному масиві

Таким чином, розмиття за Гауссом дозволяє позбутися від небажаних шумів на зображеннях, що мінімізує їх вплив на подальшу класифікацію облич.

Результат застосування фільтра Гауса до всього зображення показано на (рис. 2.4).

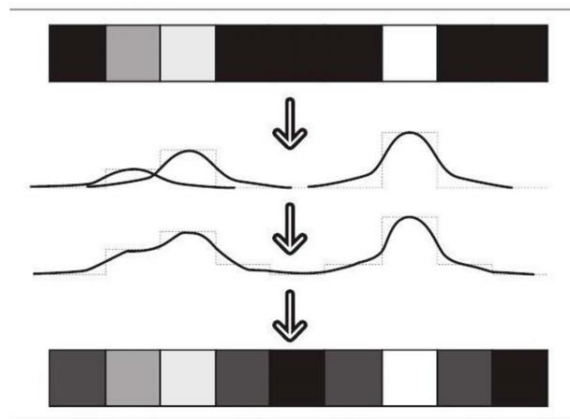


Рисунок 2.4 - Результат застосування фільтра Гауса

## 2.3 Перетворення LBSH

### 2.3.1 Класичний метод

Оператор LBSH, уперше запропонований у 1996 році для класифікації текстур, знайшов своє застосування й у задачах розпізнавання обличчя оператора полягає у використанні порогового перетворення яскравості, де значення інтенсивності центрального пікселя порівнюється з інтенсивностями сусідніх пікселів, формуючи двійковий шаблон, який кодує кожен піксель у відношенні до його оточення.

У своєму класичному варіанті LBSH застосовує матрицю  $3 \times 3$ , що охоплює сусідні пікселі навколо центрального. Це оточення дозволяє створити локальну двійкову карту на основі відносних яскравостей, а результат порівняння об'єднується у двійкове число, яке кодує текстуру навколо кожного пікселя. Такий підхід генерує унікальні двійкові "відбитки" кожної текстури, зберігаючи ключові відмінності між ними. На рисунку подано ілюстрацію обчислення цього

перетворення для шаблону розміром 3x3. (рис. 2.5)

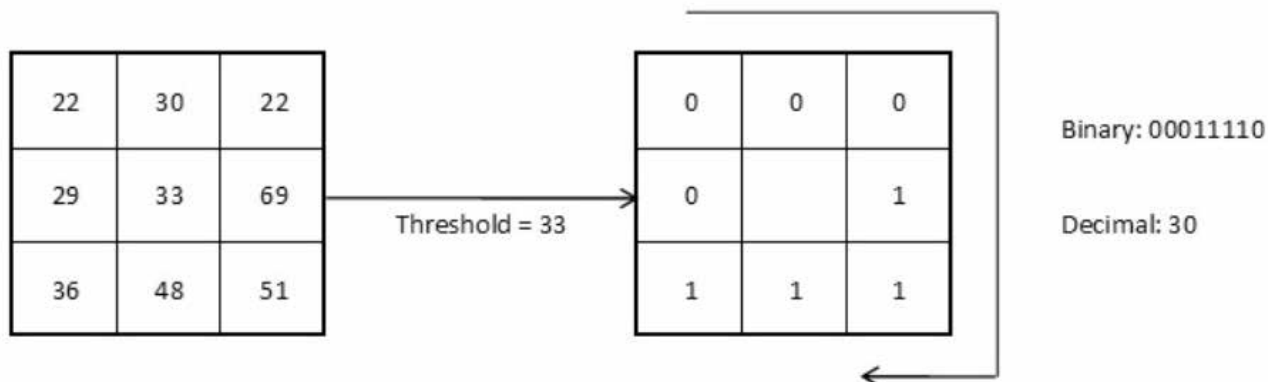


Рисунок 2.5 - Класичний оператор LBS

Для кожного прямокутника області, отриманого після застосування оператора LBSH, розраховуються гістограми, які відображають частоту зустрічі пікселів з різною яскравістю в цій області. Значення елементів гістограми LBS можна описати такою формулою:

$$H_i = \sum_{x,y} I\{f(x,y) = i\}, i = 0, \dots, n - 1 \quad (3)$$

Отримані гістограми нормалізуються, об'єднуються та використовуються пізніше як ознаки класифікації. У оригінальному дослідженні для класифікації використовується метод найближчого сусіда, який буде детально описано в наступних розділах. Приклад поділу зображення на прямокутні області та формування гістограм можна побачити на (рис. 2.6)

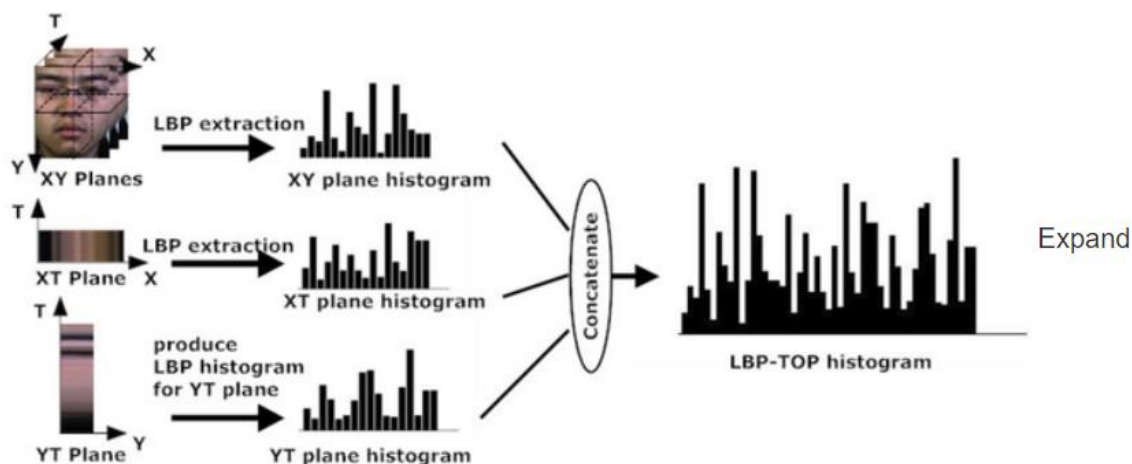


Рисунок 2.6 - Розбиття зображення на прямокутні області формування Гістограми

У результаті ми отримуємо зображення, описане на трьох рівнях локалізації. Також такий опис зображень не буде залежним від монотонних змін в освітленні на зображенні.

### 2.3.2. Уніформа ЛБШ

Дослідження показують, що серед локальних бінарних патернів (ЛБШ) лише певні шаблони несуть значущу інформацію про контури та форму об'єктів на зображенні. Було визначено, що саме так звані "рівномірні" ЛБШ є особливо корисними для аналізу — вони мають чітко структуровані властивості, які допомагають точно окреслити форму об'єктів. Під "рівномірними" шаблонами розуміють такі ЛБШ, у яких двійковий код демонструє не більше двох переходів між нулями й одиницями. Завдяки цьому, рівномірні ЛБШ зосереджуються на значних локальних характеристиках, таких як кінці ліній, грані чи яскраві точки, і водночас успішно відсіюють другорядні деталі та шум.

На рисунку 2.7 представлено приклади рівномірних ЛБШ, які ілюструють такі ключові локальні особливості.

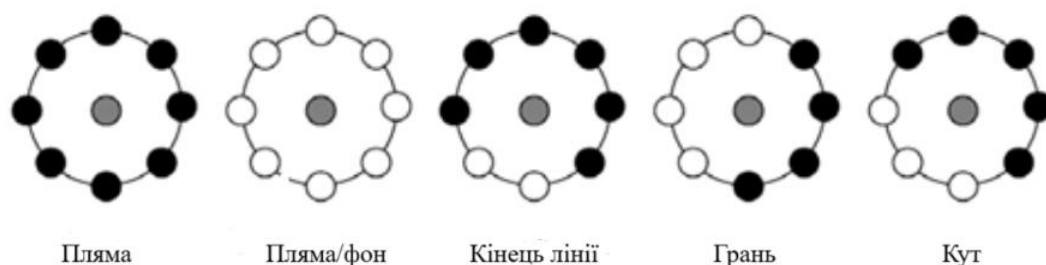


Рисунок 2.7 – Уніформа ЛБШ

У наборі рівномірних локальних бінарних шаблонів (ЛБШ) налічується 58 унікальних шаблонів. Таким чином, після додавання одного додаткового шаблону для нерівних ЛБШ, отримується 59-мірна гістограма ознак. У порівнянні з оригінальним методом, де використовувалась 256-мірна гістограма, таке зменшення розмірності дозволяє зменшити використання пам'яті.

Це досягається за рахунок використання лише важливих функцій, які представлені рівномірними ЛБШ. Застосування меншої кількості ознак дозволяє зосередитися на розпізнаванні суттєвих особливостей об'єктів і знизити вплив шуму або незначних деталей на процес класифікації. У результаті підвищується продуктивність алгоритму, забезпечуючи ефективне розпізнавання об'єктів.

### 2.3.3 Центральньо-симетрична ЛБШ

Ця модифікація методу обчислення локальних бінарних шаблонів сприяє подальшому зменшенню використання пам'яті та обчислювальної складності під час класифікації. Суть цієї модифікації полягає в тому, що граничним значенням

для кожного пікселя в межах околиці є значення яскравості пікселя, який знаходиться у протилежному положенні від центру околиці, а не значення яскравості центрального пікселя околиці [23].

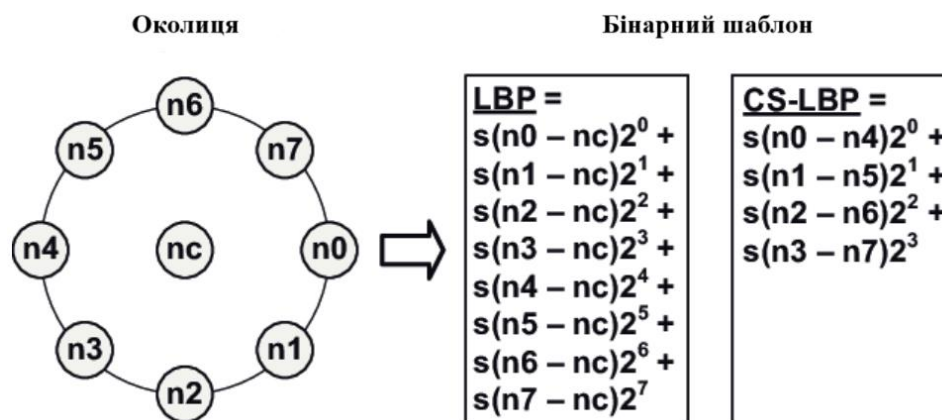


Рисунок 2.8 – Розрахунок значення умовного та центрально-симетричного

У даному методі кількість цифр значень перетворення пікселів зменшується до чотирьох, що призводить до зменшення розміру гістограми ознак до 16 ( $2^4 = 16$ ). Це дозволяє ще більше зекономити пам'ять та підвищити швидкість класифікації порівняно з однорідним ЛБШ. Такі переваги роблять цю модифікацію ідеальним вибором для класифікації в реальному часі. Проте, зменшення розмірності гістограми ознак може негативно вплинути на точність класифікації. Тому при виборі класифікаційних ознак потрібно знайти компроміс між швидкістю роботи та точністю. В даній роботі окремий розділ присвячений дослідженню продуктивності вищезгаданих методів з метою знайти оптимальне рішення.

## 2.4 Маска значущих ділянок зображення

Після виявлення облич отримані зображення мають квадратну форму, однак

обличчя займає лише частину цього простору, і на класифікацію можуть впливати сторонні, незначущі ділянки зображення. Щоб уникнути впливу цих пустих або малозначущих областей на точність результатів, виникає потреба у фільтрації активних ділянок.

Одним із найбільш раціональних способів вирішення є введення маски: вона повторює розміри зображення, проте зберігає виділеними лише ті області, що містять суттєві елементи для розпізнавання. У даному випадку ключовим об'єктом виступає овал в центрі кадру, який обмежує зону класифікації лише контуром обличчя. Такий підхід не тільки усуває фонові шуми та незначущі деталі, але й підвищує точність алгоритму, дозволяючи йому зосередитися виключно на релевантних характеристиках обличчя, що помітно покращує результати класифікації.

Значення пікселів отриманого зображення визначаються за формулою:

$$R(x, y) = \begin{cases} I(x, y) & \text{якщо } M(x, y) \neq 0 \\ 0 & \text{якщо } M(x, y) = 0 \end{cases} \quad (4)$$

де  $I(x, y)$  – значення яскравості пікселя обробленого зображення;

$M(x, y)$  – значення яскравості пікселів маски значущих областей.

У процесі класифікації, що використовує локальні бінарні шаблони (LBS), важливим кроком є застосування маски значущих областей після LBS-перетворення, але до обчислення гістограм. Такий підхід дає змогу об'єднати всі незначущі пікселі в один категорійний елемент на гістограмі, суттєво знижуючи шум, що надходить до класифікатора.

На (рис. 2.10) проілюстровано ефект цієї маски, де завдяки фокусуванню на значущих областях, кожен важливий піксель утворює інформативний патерн, тоді як незначущі регіони консолідуються в один елемент. Така маска дозволяє системі розпізнавання виділяти справді критичні елементи та істотно підвищує чіткість гістограмних характеристик, відокремлюючи їх від "фонових" значень.



Рисунок 2.10 - Нанесення маски значних площ.

## 2.5 Метод найближчого сусіда

Після побудови гістограм LBS, класифікація виконується через метод найближчого сусіда, як це було вперше застосовано у базовому дослідженні LBS для розпізнавання обличчя [14]. Суть цього методу полягає в тому, що об'єкт приписується до класу, до якого належить найближчий елемент за відстанню. На ілюстрації (рис. 2.11) жовтий квадрат, відповідно до цього правила, потрапляє до класу зелених трикутників.

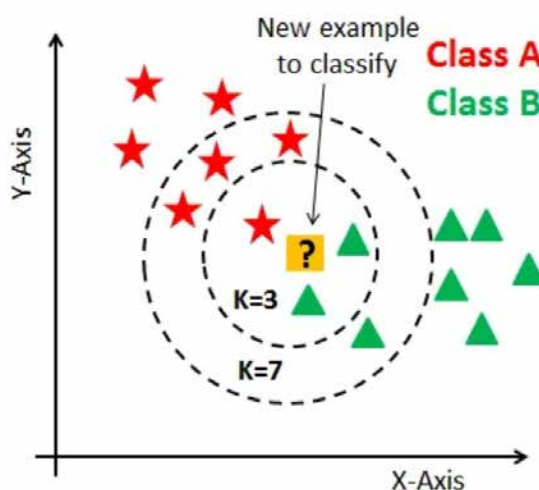


Рисунок 2.11 – Метод найближчого сусіда

Для покращення точності іноді застосовують підхід, де об'єкт класифікується відповідно до переважної більшості сусідів у заданому радіусі. Проте експерименти, проведені в цій роботі, виявили, що для розпізнавання обличчя цей метод не призводить до очікуваного підвищення точності. Ба більше, врахування більшості сусідів може навіть знижувати ефективність, оскільки значно підвищує вплив шуму в зображенні.

Метод найближчого сусіда, незважаючи на свою простоту, вважається надзвичайно ефективним у багатьох завданнях класифікації. Проте його вразливість до неправильних або аномальних значень даних лишається його основним недоліком. Тому, хоча він забезпечує високу продуктивність для різноманітних задач, у випадках, коли дані містять значну кількість шуму або помилок, необхідно обережно застосовувати підхід із більшістю сусідів, щоб уникнути погіршення результатів.

## **2.6 Планування експериментальних досліджень**

З огляду на вимоги швидкості, центрально-симетричні локальні бінарні шаблони постають ідеальним вибором для реальних часових систем розпізнавання завдяки їх обчислювальній ефективності. Однак перед їхньою інтеграцією до кінцевої системи варто ретельно оцінити їхню продуктивність у порівнянні з іншими, хоча й повільнішими, варіаціями ЛБШ для точності класифікації.

Для розв'язання цієї дилеми було обрано й протестовано всі три методи обчислення гістограми ЛБШ на двох різних наборах даних. Водночас здійснювався підбір оптимальної стратегії розбиття зображень на локальні області. Всі тести проводилися за допомогою методу найближчого сусіда для класифікації, а реалізації ЛБШ і самого методу найближчого сусіда, створені в ході дослідження, лягли в основу системи розпізнавання осіб у відеопотоках. Також проводилося порівняння швидкості роботи системи з різними варіантами перетворення ЛБШ.

Серед використаних для тестування наборів був набір зображень облич, наданий Кембриджською лабораторією [25], що включає 40 осіб, кожна з яких представлена 10 зображеннями. Цей набір цікавий тим, що на ньому зображені різні позиції кожної особи за стабільного освітлення. На (рис. 2.12) наведено приклад зображення з цього набору для однієї з осіб.



Рисунок 2.12 – Обличчя Оліветті

Як навчальні та тестові зразки використовували по 5 зображень для кожної людини. Тестова і навчальна вибірки не мали перетину, але для отримання результатів на більшому наборі даних тестову і навчальну вибірки поміняли місцями і тестування повторили. Перед обробкою зображення було масштабовано до розміру 128\*128 пікселів. В результаті всього було класифіковано 400 зображень.

Другий набір даних — це база даних зображень обличчя лабораторії Єльського університету [26]. Ця база містить зображення 38 людей, по 65 зображень на кожного, включаючи різні зображення з освітленням. З них для тестування було відібрано по 10 зображень для кожної людини. Приклад зображень з другої основи граней наведено на (рис. 2.13).

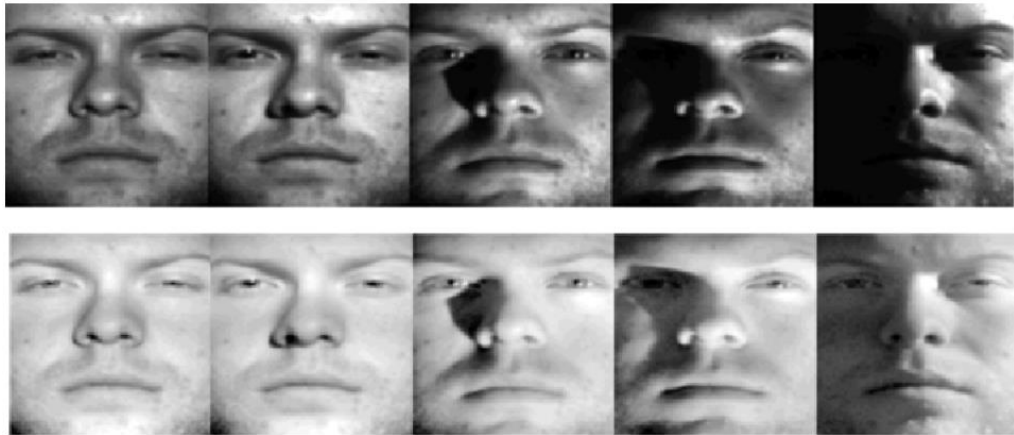


Рисунок 2.13 – Зображення з бази даних Єльського університету

Важливо зазначити, що в цій базі даних обличчя були кадровані більш точно і в більшому масштабі, ніж на зображеннях першого набору даних. Тестування проводилося у подібний спосіб до першого набору. Загалом було класифіковано 380 зображень

## 2.7 Висновки з розділу 2

У другому розділі представлено детальний розгляд алгоритму виявлення обличчя методом Віоли-Джонса, розкрито його ключові етапи та взаємозв'язки. Зокрема, розглянуто застосування Гаусового фільтра як засобу для приглушення шумів у вхідних зображеннях, що досягається шляхом розмиття, мінімізуючи шумовий вплив на точність класифікації.

Крім того, розкрито тонкощі роботи оператора локальних бінарних шаблонів (LBSH) та його вдосконалених модифікацій. Описано, як саме обчислюються гістограми LBS, що слугують інструментом для виділення значущих локальних ознак і детального розпізнавання текстури на рівні пікселів.

Далі розглядається застосування маски для фільтрації зображення — маска виключає ділянки, які не містять обличчя, акцентуючи увагу на важливих площах, що значно підвищує загальну точність розпізнавання. На завершення наведено метод класифікації отриманих гістограм за допомогою алгоритму найближчого сусіда, з акцентом на обчисленні відстаней між гістограмами для визначення схожості об'єктів, що є ключовим етапом у порівнянні й розрізненні зразків.

### **3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ**

#### **3.1 Інструментарій для розробки програмного забезпечення**

Для розробки цієї програми було вибрано мову Python, середовище PyCharm і бібліотеку OpenCV, що спільно забезпечують як легкість у реалізації, так і високу продуктивність при виконанні завдань розпізнавання облич.

Python — це високорівнева, інтерпретована мова програмування. Завдяки лаконічному та зрозумілому синтаксису Python дозволяє швидко створювати прототипи та експериментувати з параметрами — це особливо важливо в задачах комп'ютерного зору, де час має вирішальне значення. Окрім простоти, Python також має розгалужену екосистему бібліотек для комп'ютерного зору, зокрема OpenCV, NumPy та Scikit-learn, що сприяє більшій ефективності. Спільнота користувачів Python у галузі штучного інтелекту також забезпечує доступ до великої бази знань, дозволяючи швидко вирішувати проблеми завдяки готовим прикладам і порадам. Крім того, Python працює на різних операційних системах, що робить його універсальним інструментом для розробки.

Python має переваги в роботі з нейронними мережами через свій високий рівень абстракції та простий синтаксис, що дозволяє зосередитись на алгоритмах і

архітектурі нейронної мережі, а не на деталях реалізації. Це зручно для швидкого створення і тестування складних моделей, оскільки дозволяє писати код у зрозумілій формі, наближеній до математичного опису. Його інтерпретована природа спрощує ітерації, що є важливим у процесі побудови, відлагодження та навчання моделей, де параметри часто коригуються вручну.

Також Python добре справляється з паралельними обчисленнями, які критично важливі для тренування великих нейронних мереж. Завдяки зручному доступу до GPU (через бібліотеки, які використовують можливості мови, наприклад CUDA), Python прискорює навчання та масштабування мереж.

PyCharm — це потужне IDE, яке пропонує безліч інструментів для розробки на Python, включно з автодоповненням коду, синтаксичним аналізом і підтримкою різних версій Python. Це дозволяє організовувати великі проєкти, керувати середовищем і оновлювати бібліотеки. Вбудований налагоджувач PyCharm полегшує відстеження помилок і має підтримку плагінів, що розширюють його функціонал і дають змогу інтегрувати додаткові інструменти, такі як бази даних і Git, роблячи розробку максимально зручною. (рис. 3.1)

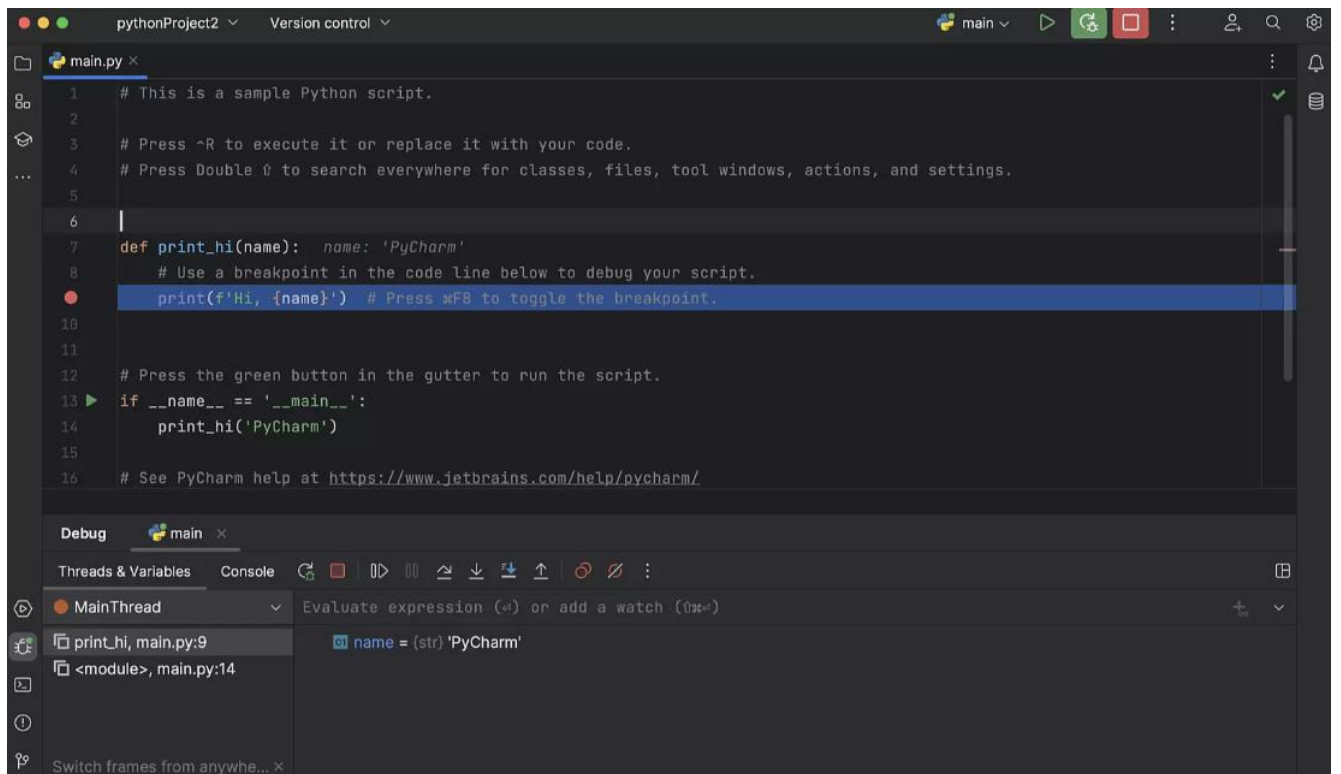


Рисунок 3.1 – Інтерфейс програми PyCharm Community edition

Бібліотека OpenCV (Open Source Computer Vision Library) спеціалізується на комп'ютерному зорі й містить широкий спектр алгоритмів для обробки зображень і відео, зокрема функції детекції обличчя. Завдяки вбудованим алгоритмам, як-от LBPН (Local Binary Patterns Histograms), OpenCV дозволяє досягти відмінної точності навіть із відносно малими датасетами. OpenCV, написаний на С++ і оптимізований для високої продуктивності, забезпечує обробку зображень із високою швидкістю, що важливо для програм розпізнавання в реальному часі. OpenCV стала стандартом в галузі комп'ютерного зору завдяки своїй універсальності та потужному функціоналу. До її основних переваг можна віднести:

- Відкритий код і безкоштовність
- Широкий функціонал
- Підтримка багатьох мов програмування
- Велика спільнота

- Кросплатформність
- Велика кількість попередньо реалізованих алгоритмів
- Інтеграція з іншими бібліотеками

Недоліки:

- Споживання системних ресурсів
- Обмежена підтримка деяких алгоритмів
- Для повного освоєння всього функціоналу потрібно багато часу

Отже, комбінація Python, PyCharm і OpenCV дає змогу швидко створити, налагодити й оптимізувати систему розпізнавання облич із чудовою точністю й продуктивністю. Python із його різноманітними бібліотеками, зручне середовище розробки PyCharm і продуктивність OpenCV забезпечують максимальну гнучкість та ефективність, необхідні для виконання поставленого завдання.

### 3.2 Структура проекту

Структура проекту налічує 2 папки та 5 файлів.

Venv - це директорія, де створюється віртуальне середовище Python, яке містить усі специфічні бібліотеки та залежності, необхідні для роботи конкретного проекту. Віртуальне середовище діє як захисний шар між глобальними залежностями і проектом, ізолюючи їх, що допомагає уникнути конфліктів версій бібліотек, зберігаючи стабільність та передбачуваність у кожному окремому проекті.

Datasets - ця папка слугує сховищем для даних, необхідних для роботи системи розпізнавання облич. У ній зберігаються набори даних, які застосовуються для збору зображень, тренування та тестування моделі. Вона є критично важливим елементом у підготовці та управлінні даними для якісного навчання.

datacollect.py - це сценарій, що призначений для збору зображень облич, який допомагає формувати базовий набір даних. Під час роботи він активує камеру,

застосовує метод каскаду Хаара для виявлення обличч через файл `haarcascade_frontalface_default.xml`, та зберігає зображення для подальшого використання в тренуванні моделі.

`haarcascade_frontalface_default.xml` – це файл XML, в якому збережено попередньо навчений каскад Хаара, спеціально налаштований для детекції обличч. Використовується OpenCV для швидкого розпізнавання обличч у відеопотоці або на окремих зображеннях. Цей файл є основним інструментом для точного виявлення обличч під час збору та тестування даних.

`testmodel.py` - цей скрипт відповідає за тестування вже натренованої моделі розпізнавання обличч. Він завантажує модель з файлу `Trainer.yml`, обробляє вхідні зображення або відеопотік і виконує розпізнавання обличч, відображаючи ім'я розпізнаної особи або позначку "Unknown" для тих, кого не розпізнано.

`Trainer.yml` – це файл, де зберігається тренувана модель розпізнавання обличч, створена за допомогою алгоритму LBP. Після виконання скрипту `trainingdemo.py` він містить деталі про обличчя, з якими модель була навчена. Цей файл використовується на етапі тестування та розпізнавання для швидкого доступу до навченої моделі.

`trainingdemo.py` – скрипт, що призначений для тренування моделі розпізнавання обличч. Він зчитує зображення обличч із директорії `datasets`, обробляє їх і навчає модель, яка зберігається у файлі `Trainer.yml` для подальшого застосування.

У сукупності ці файли утворюють повний цикл для системи розпізнавання обличч: від збирання даних, тренування моделі до її тестування та реального розпізнавання.

### **3.3 Datacollect.py**

`Datacollect.py` це файл, що призначений для збору зображень обличч, який допомагає формувати базовий набір даних. Під час роботи він активує камеру,

застосовує метод каскаду Хаара для виявлення обличч через файл `haarcascade_frontalface_default.xml`, та зберігає зображення для подальшого використання в тренуванні моделі. Код програми та опис кожної функції:

`import cv2`: Імпортує бібліотеку OpenCV, яка використовується для обробки зображень та роботи з комп'ютерним зором.

`from cv2.cv2 import CascadeClassifier`: Імпортує клас `CascadeClassifier`, який використовується для завантаження і використання каскадного класифікатора (наприклад, каскад Хаара), що допомагає виявляти обличчя на зображенні.

`video = cv2.VideoCapture(1)`: Ініціалізує відеозапис з вебкамери. Параметр 1 означає, що програма намагається використовувати другу підключену камеру (індексація камер починається з 0).

`facetedect = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")`: Створює об'єкт `CascadeClassifier`, завантажуючи в нього каскад Хаара з файлу `haarcascade_frontalface_default.xml`. Цей файл містить попередньо навчену модель для виявлення обличч на зображенні.

`id = input("Enter your id ")`: Запитує ID користувача. Це ID буде використано для іменування зображень у наборі даних, щоб позначити, якому користувачу належать збережені обличчя.

`count = 0`: Ініціалізує лічильник, який використовується для підрахунку збережених зображень обличчя.

`while True::` Початок безкінечного циклу для обробки кожного кадру з відео.

`ret, frame = video.read()`: Захоплює один кадр з камери. Змінна `ret` показує, чи захоплення кадру було успішним, а `frame` містить сам кадр.

`gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`: Перетворює кадр на відтінки сірого (градації сірого), оскільки алгоритм детекції обличч працює ефективніше з чорно-білими зображеннями.

`faces = facetedect.detectMultiScale(gray, 1.3, 5)`: Використовує `CascadeClassifier` для виявлення обличч.

`for (x, y, w, h) in faces::` Перебирає кожне виявлене обличчя, де `(x, y)` –

координати верхнього лівого кута обличчя, а  $w$  і  $h$  – ширина та висота прямокутника навколо обличчя.

`count = count + 1`: Збільшує лічильник зображень облич, щоб уникнути дублювання імен файлів.

`cv2.imwrite('datasets/User.' + str(id) + "." + str(count) + ".jpg", gray[y:y+h, x:x+w])`: Зберігає обличчя, виділене зображенням, у форматі .jpg у папці datasets. Ім'я файлу містить ID користувача та поточний номер зображення (лічильник).

`cv2.rectangle(frame, (x, y), (x+w, y+h), (50, 50, 255), 1)`: Малює червоний прямокутник навколо виявленого обличчя на кадрі для візуалізації.

`cv2.imshow("Frame", frame)`: Відображає кадр з обрамленим обличчям у вікні з назвою "Frame".

`k = cv2.waitKey(1)`: Затримка на 1 мс між кадрами і очікування натискання клавіші.

`if count > 1000::` Перевіряє, чи зібрано понад 1000 зображень обличчя. Якщо так, цикл завершується.

Блок-схема роботи програмного коду для збору збору зображень для подальшого тренування нейронної мережі для розпізнавання обличчя людини. (Рис. 3.2)



Рисунок 3.2 – Збір зображень

### 3.4 Trainingdemo.py

Скрипт, що призначений для тренування моделі розпізнавання облич. Він зчитує зображення облич із директорії datasets, обробляє їх і навчає модель, яка

зберігається у файлі `Trainer.yml` для подальшого застосування.

У сукупності ці файли утворюють повний цикл для системи розпізнавання облич: від збирання даних, тренування моделі до її тестування та розпізнавання.

`import cv2`: Імпортує бібліотеку `OpenCV` для роботи з комп'ютерним зором.

`import numpy as np`: Імпортує бібліотеку `NumPy`, яка використовується для роботи з багатовимірними масивами та операціями над ними.

`from PIL import Image`: Імпортує модуль `Image` з бібліотеки `Pillow (PIL)` для роботи із зображеннями, дозволяючи їх відкрити, обробити та конвертувати.

`cv2.face.LBPHFaceRecognizer_create()`: Створює екземпляр розпізнавача облич на основі алгоритму `LBPH (Local Binary Patterns Histogram)`.

Параметри `radius`, `neighbors`, `grid_x`, `grid_y` налаштовують розпізнавач:

`radius=2`: Радіус для формування локальних бінарних шаблонів. Визначає розмір області навколо кожного пікселя.

`neighbors=8`: Кількість сусідніх пікселів для порівняння. Вказує на кількість пікселів, які використовуються для створення шаблону.

`grid_x=8` і `grid_y=8`: Розмір сітки для обчислення гістограм, що збільшує деталізацію в залежності від розмірів обличчя.

`path = "datasets"`: Вказує шлях до папки з набором даних, що містить зображення облич для тренування.

Розбір функції `getImageID`

`imagePath = [os.path.join(path, f) for f in os.listdir(path)]`: Створює список шляхів до всіх файлів у папці `path`, використовуючи `os.path.join` для коректного поєднання шляху та імені файлу.

`faces = []` і `ids = []`: Ініціалізує два порожніх списки: `faces` для зберігання зображень облич у вигляді масивів. `ids` для зберігання ID кожного обличчя.

`for imagePath in imagePath::` Проходить по кожному шляху до зображення в списку `imagePath`.

`faceImage = Image.open(imagePaths).convert('L')`: Відкриває зображення за допомогою `Image.open()` з бібліотеки `PIL` і конвертує його в градації сірого ('L'), що

спрощує обробку зображень.

`faceNP = np.array(faceImage)`: Перетворює зображення обличчя в об'єкт NumPy (масив), щоб зробити його придатним для роботи з OpenCV.

`Id = int(os.path.split(imagePaths)[-1].split(".")[1])`: Витягує ID з імені файлу

`os.path.split(imagePaths)[-1]`: Отримує ім'я файлу з повного шляху.

`.split(".")[1]`: Розділяє ім'я файлу за крапками і вибирає частину з ID.

`int(...)`: Конвертує ID в ціле число.

`faces.append(faceNP)` і `ids.append(Id)`: Додає масив зображення `faceNP` до списку `faces` і ID `Id` до списку `ids`.

`cv2.imshow("Training", faceNP)`: Відображає поточне зображення обличчя під час тренування в окремому вікні з назвою "Training".

`cv2.waitKey(1)`: Чекає 1 мс, щоб дати можливість OpenCV відобразити зображення в реальному часі.

`return ids, faces`: Повертає два списки: `ids` (список ID) і `faces` (список облич у вигляді масивів).

Отримання даних та тренування розпізнавача

`IDs, facedata = getImageID(path)`: Викликає функцію `getImageID`, щоб отримати списки `IDs` та `facedata`, які містять відповідно ID та масиви облич.

`recognizer.train(facedata, np.array(IDs))`: Використовує метод `train` для навчання моделі LBPН розпізнавання облич, передаючи:

`facedata` — масив зображень облич.

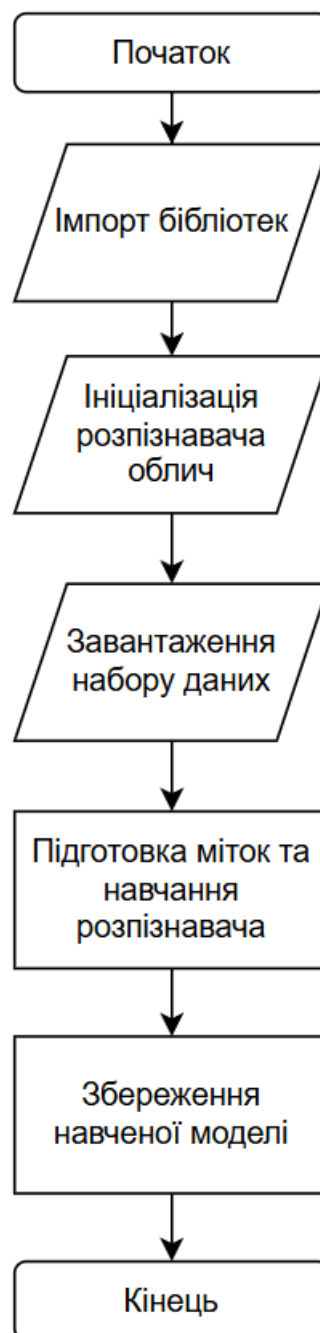
`np.array(IDs)` — масив ID, що відповідає кожному зображенню.

`recognizer.write("Trainer.yml")`: Зберігає навчений розпізнавач у файл `Trainer.yml`. Цей файл містить модель, яку можна завантажити в інші програми для розпізнавання облич на основі даних, що були зібрані та навчені.

`cv2.destroyAllWindows()`: Закриває всі вікна, відкриті OpenCV під час тренування.

`print("Training Complete.")`: Виводить повідомлення про завершення процесу навчання.

Блок-схема роботи програмного коду для тренування моделі розпізнавання облич, що використовує набір даних отриманий після виконання Trainingdemo.py. (Рис. 3.3)



## Рисунок 3.3 – Збір зображень

### 3.5 Testmodel

`import cv2`: Імпортує бібліотеку OpenCV, яка використовується для обробки зображень і роботи з комп'ютерним зором.

`from cv2.cv2 import CascadeClassifier`: Імпортує клас CascadeClassifier із OpenCV, який дозволяє використовувати методи каскадного класифікатора для виявлення об'єктів, зокрема облич.

`cv2.VideoCapture(1)`: Створює об'єкт для захоплення відео з камери. Аргумент 1 вказує на використання другої підключеної камери (якщо 0 — це перша камера, 1 — друга, і т.д.).

`cv2.CascadeClassifier("haarcascade_frontalface_default.xml")`: Створює об'єкт facedetect для виявлення обличчя, завантажуючи попередньо навчений каскадний класифікатор з файлу haarcascade\_frontalface\_default.xml. Цей файл містить модель, яка використовується для виявлення облич у кадрі.

`cv2.face.LBPHFaceRecognizer_create()`: Створює об'єкт розпізнавача облич на основі LBPH (Local Binary Patterns Histograms).

Параметри:

`radius=2`: Визначає радіус для обчислення локальних бінарних шаблонів.

`neighbors=8`: Визначає кількість сусідів для аналізу.

`grid_x=8` і `grid_y=8`: Визначає кількість гістограм, що будуть створені по горизонталі і вертикалі.

`recognizer.read("Trainer.yml")`: Завантажує попередньо навчений файл моделі Trainer.yml, який містить дані для розпізнавання облич.

`name_list`: Список імен, які відповідають ID користувачів. Перший елемент

пустий, оскільки ідентифікація зазвичай починається з ID 1 (наприклад, "Denys" може мати ID 1, а "Zheka" — ID 2).

`ret, frame = video.read()`: Захоплює кадр з відеопотоку. `frame` містить зображення, а `ret` — булеве значення, яке вказує, чи успішно захоплено кадр.

`gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`: Конвертує кадр в градації сірого для полегшення обробки, оскільки детекція обличчя працює швидше і точніше з чорно-білим зображенням.

`faces = facedetect.detectMultiScale(gray, 1.3, 5)`: Виконує виявлення облич у кадрі.

1.3: Коефіцієнт масштабу, який визначає зменшення розміру зображення при кожному проході.

5: Мінімальна кількість сусідніх об'єктів, які повинні бути виявлені, щоб вважати обличчя достовірно знайденим.

`for (x, y, w, h) in faces::` Цикл проходить по координатах і розмірах кожного виявленого обличчя.

`serial, conf = recognizer.predict(gray[y:y+h, x:x+w])`: Використовує розпізнавач для прогнозування ID та рівня достовірності `conf` для обличчя, вирізаного з сірого зображення `gray[y:y+h, x:x+w]`.

`if conf > 80::` Перевіряє рівень достовірності розпізнавання. Якщо `conf` (рівень довіри) менше 50, результат вважається достовірним.

`cv2.putText(frame, name_list[serial], (x, y - 46), cv2.FONT_HERSHEY_SIMPLEX, 1, (50, 50, 255), 2)`: Виводить ім'я, пов'язане з `serial` (ID), над обличчям.

`cv2.rectangle(frame, (x, y), (x + w, y + h), (50, 50, 255), 1)`: Малює прямокутник навколо облич

Інша гілка `else`: Якщо рівень довіри `conf` більше або дорівнює 50, обличчя позначається як "Unknown".

`k = cv2.waitKey(1)`: Чекає 1 мс на натискання клавіші.

`if k == ord("q")::` Якщо натиснута клавіша `q`, виходить із циклу, що завершує

програму.

`video.release()`: Звільняє ресурс відеозахоплення, закриваючи відеопотік.

`cv2.destroyAllWindows()`: Закриває всі вікна OpenCV, які були відкриті.

`print("Dataset Collection Done")`: Виводить повідомлення "Dataset Collection Done" після завершення програми.

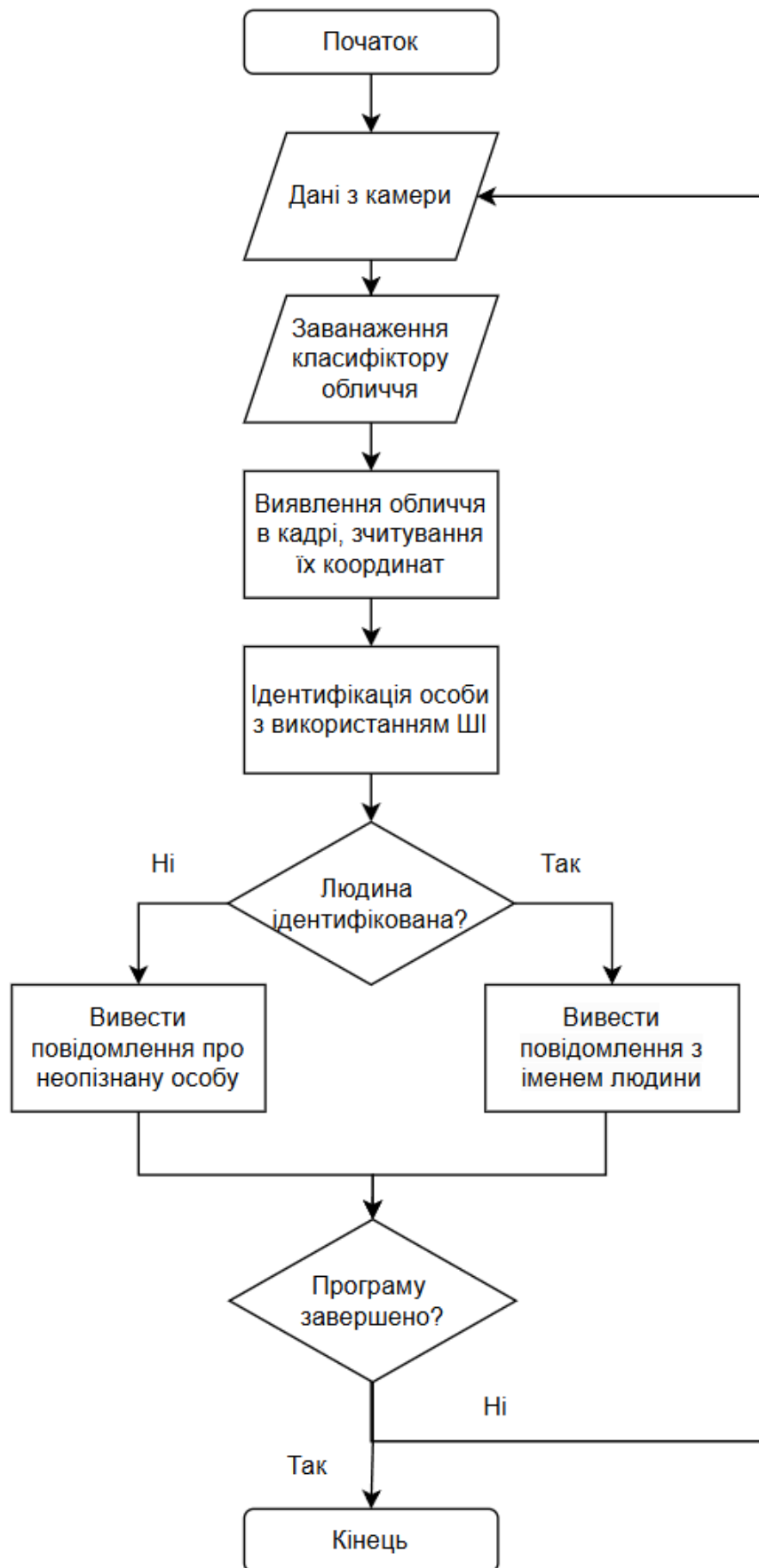


Рисунок 3.4 – Блок-схема роботи системи з розпізнаванням обличчя

### 3.6 Опис інтерфейсу комп'ютерної системи

При проектуванні програми акцент був зроблений на лаконічності, простоті та функціональності. Інтерфейс програми складається з вікна розміром 1280 на 720 пікселів. При старті програмний код ініціалізує відеопотік з веб-камери ноутбука і виводить його у наш фрейм. Далі це зображення за допомогою функцій перетворюється на чорно-білий формат та передається класифікатору, який, використовуючи свій функціонал, знаходить на відео обличчя людини. Потім це обличчя виділяється квадратом розміром 50 на 50 пікселів. Наступним кроком до знайденого обличчя застосовується наша натренована нейронна мережа, яка порівнює характерні ознаки рис обличчя і співставляє їх із наявною у нас базою даних користувачів. При вдалому розпізнаванні біля рамки виводиться ім'я людини, в іншому ж випадку з'являється напис "Unknown". (рис. 3.5)

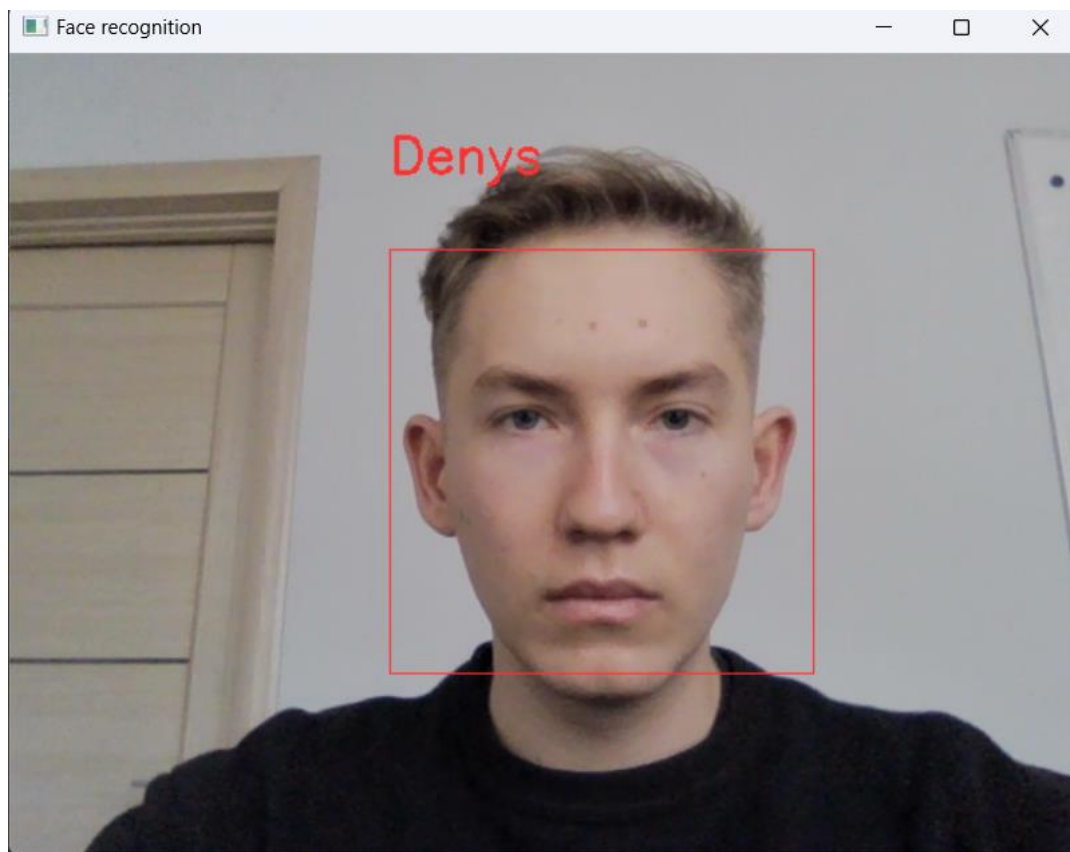


Рисунок 3.5 – Блок-схема роботи системи з розпізнаванням обличчя

### **3.7 Висновки до розділу 3**

У третьому розділі роботи проведено детальний аналіз обраного способу реалізації комп'ютерної системи з обґрунтуванням застосованих інструментів розробки. Розглянуто особливості мови програмування Python, включаючи її переваги для розв'язання задач комп'ютерного зору та мінімізації складності коду, а також описано обмеження, що можуть вплинути на продуктивність. Описано середовище Pycharm Community Edition, його можливості для розробки та налагодження коду, а також ключову роль бібліотеки OpenCV у забезпеченні необхідних функцій комп'ютерного зору.

Розділ містить структурний опис програмного коду, включаючи основні функції, принципи їхньої роботи та використання в різних частинах програми. Для кожної функції розроблено блок-схему, що пояснює логіку та послідовність дій, забезпечуючи наочність та полегшене розуміння структури програми. Завершує розділ опис графічного інтерфейсу користувача.

## **4 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ**

У цьому розділі буде надано опис процесу тестування реалізованого класифікатора обличчя. Будуть розглянуті критерії тестування, включаючи точність розпізнавання, чутливість до змін освітлення та положення обличчя, а також швидкість роботи системи.

Це дослідження дозволить оцінити ефективність розробленої системи та зробити висновки щодо застосування штучного інтелекту у вирішенні проблем комп'ютерного зору і застосування у сферах безпеки.

Також буде проведено повне тестування усіх частин програми та описано сценарії їх застосування.

### **4.1 Збір даних**

При застосуванні штучного інтелекту у сфері комп'ютерного зору, зокрема для розпізнавання обличчя людини, дуже велику роль відіграють дані. Нейронні мережі навчаються на основі зображень обличчя, і якість цих даних безпосередньо впливає на точність розпізнавання. Мережі повинні бачити широкий спектр зображень, щоб навчитись розрізняти унікальні риси обличчя кожної людини та узагальнювати знання, щоб мати змогу розпізнати обличчя навіть у складних умовах. Для ефективного розпізнавання обличчя дуже важливо, щоб набір даних містив зображення обличчя у різних умовах: під різними кутами, з різним освітленням, виразами обличчя, та навіть у різних погодних умовах. Це дозволяє

нейронній мережі стати стійкою до варіацій і забезпечує точність розпізнавання в реальних умовах.

При реалізації моєї комп'ютерної системи для збору даних було створено окремий модуль. Під час запуску він запитує у користувача ID (рис. 4.1). Далі запускається цикл, у якому з веб-камери за допомогою класифікатора обличчя у певному проміжку часу робиться знімок обличчя людини, який зберігається під її ID (рис. 4.2). Так збираються дані для подальшого тренування нейронної мережі, яка буде використовуватися для розпізнавання та ідентифікації людини.

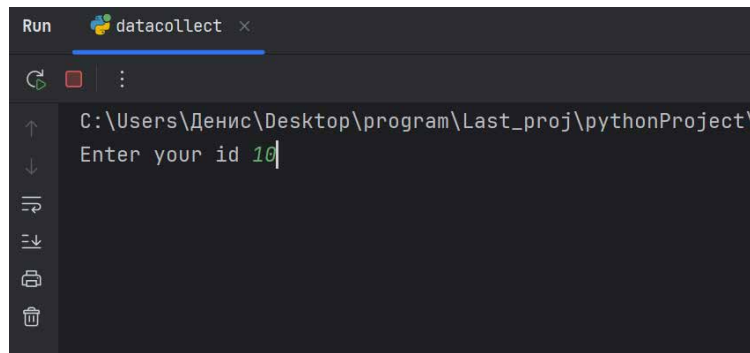


Рисунок 4.1 – Запит ід користувача

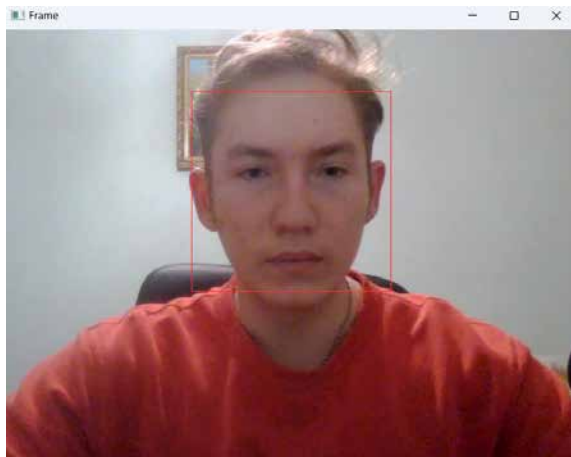


Рисунок 4.2 – Збір кадрів

У результат відпрацювання цього модулю ми отримаємо 1000 кадрів обличчя

людини, які в подальшому будуть використані у наступному модулі trainingdemo.  
(рис.4.3, 4.4, 4.5)



Рисунок 4.3 – Кадри обличчя зібрані за допомогою модуля datacollect

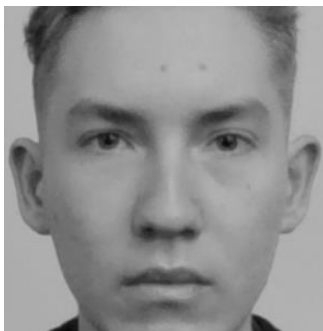


Рисунок 4.4 – Кадри обличчя зібрані за допомогою модуля datacollect



Рисунок 4.5 – Кадри обличчя зібрані за допомогою модуля datacollect

## 4.2 Тренування

За допомогою отриманого на попередніх етапах набору даних наша модель нейронної мережі починає своє навчання. Програма зчитує кожне зображення, перетворює його на градації сірого, витягує ID кожного обличчя з імені файлу та зберігає відповідні зображення і мітки в масивах для тренування. (рис. 4.6 – 4.7) Потім нейронна мережа з LBPН навчається на цих даних та зберігається у файл `Trainer.xml`, який містить всі параметри моделі для подальшого використання в розпізнаванні облич у реальному часі.

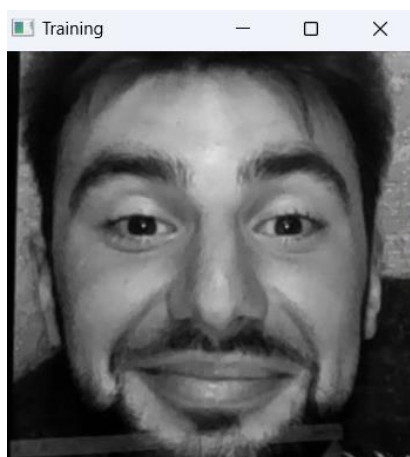


Рисунок 4.6 – Процес тренування

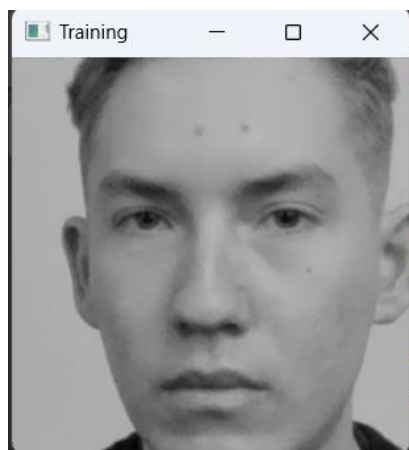


Рисунок 4.7 – Процес тренування

### 4.3 Тестуванні комп'ютерної системи

Після збору даних і тренування нейронної мережі основна програма готова до повноцінного використання. Під час запуску ініціалізується бібліотека OpenCV (cv2), яка використовується для обробки зображень і роботи з комп'ютерним зором. Клас CascadeClassifier із цієї бібліотеки дозволяє застосовувати методи каскадного класифікатора для виявлення об'єктів, зокрема облич.

Наступним кроком створюється об'єкт розпізнавача облич на основі LBPН (Локальних бінарних гістограм) і завантажується натренована модель Trainer.yml.

Застосовуючи ці інструменти, програма виконує завдання виявлення облич із веб-камери та ідентифікації людини з наявного датасету. При коректному розпізнаванні програма виділить обличчя людини та покаже ім'я, що відповідає власнику. (рис. 4.8). Коли програма не може ідентифікувати людину з відеопотоку вона відмітить обличчя як неопізнане. (рис. 4.9)

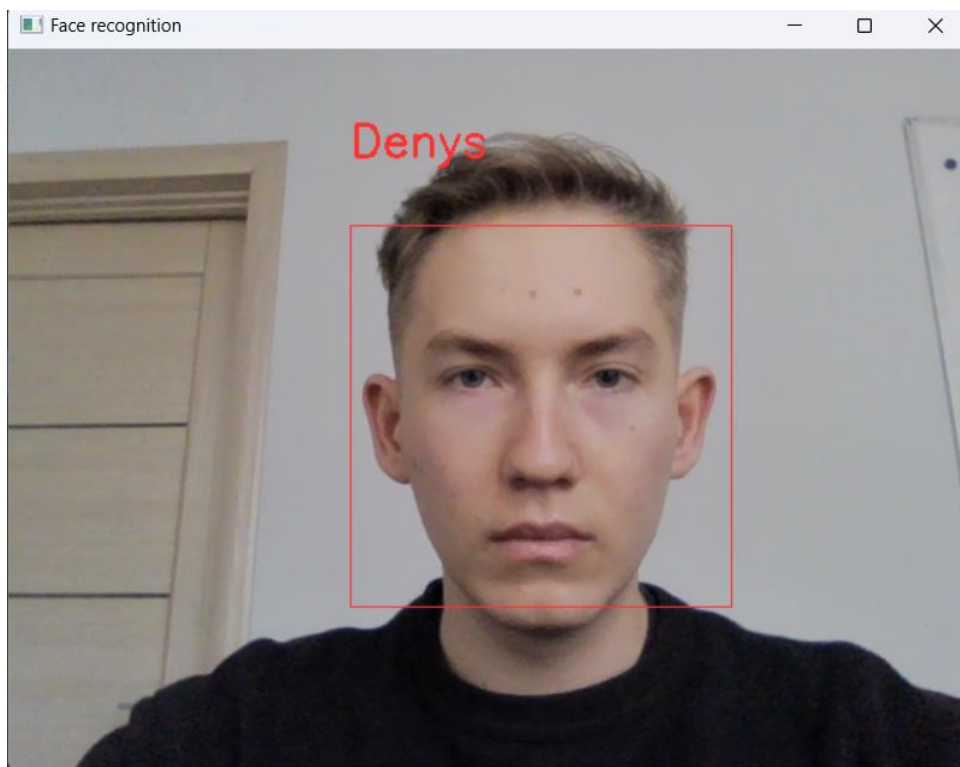


Рисунок 4.8 – Вдале розпізнавання обличчя людини

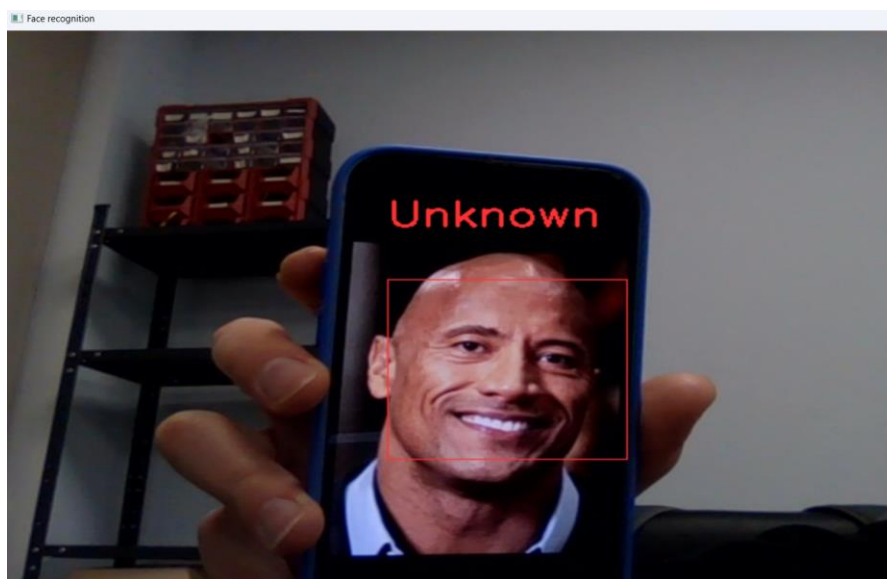


Рисунок 4.8 – Людину не опізнано

Окрема увага була приділена тестуванню системи при частковому прикритті обличчя. (рис. 4.9). Але при закриванні обличчя більше ніж на 30% система починала збоїти.(рис. 4.10) Переставав працювати детектор виявлення обличчя в

кадрі і відповідно не можливо було ідентифікувати особу, оскільки не було зразка для порівняння. При використанні капелюха чи окулярів у більшості випадків розпізнавання відбувається коректно. (Рис. 4.11)

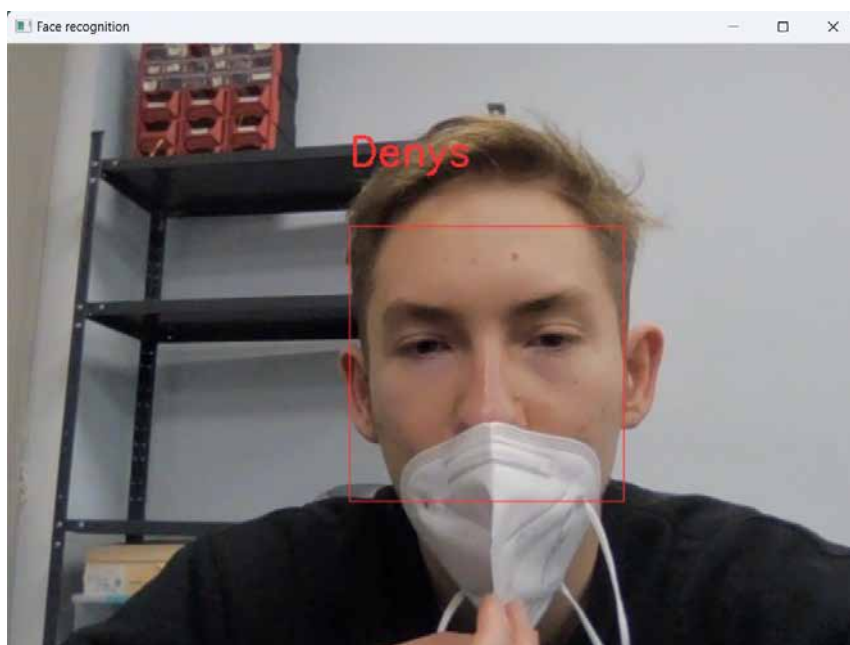


Рисунок 4.9 – Відпрацювання при закритті обличчя

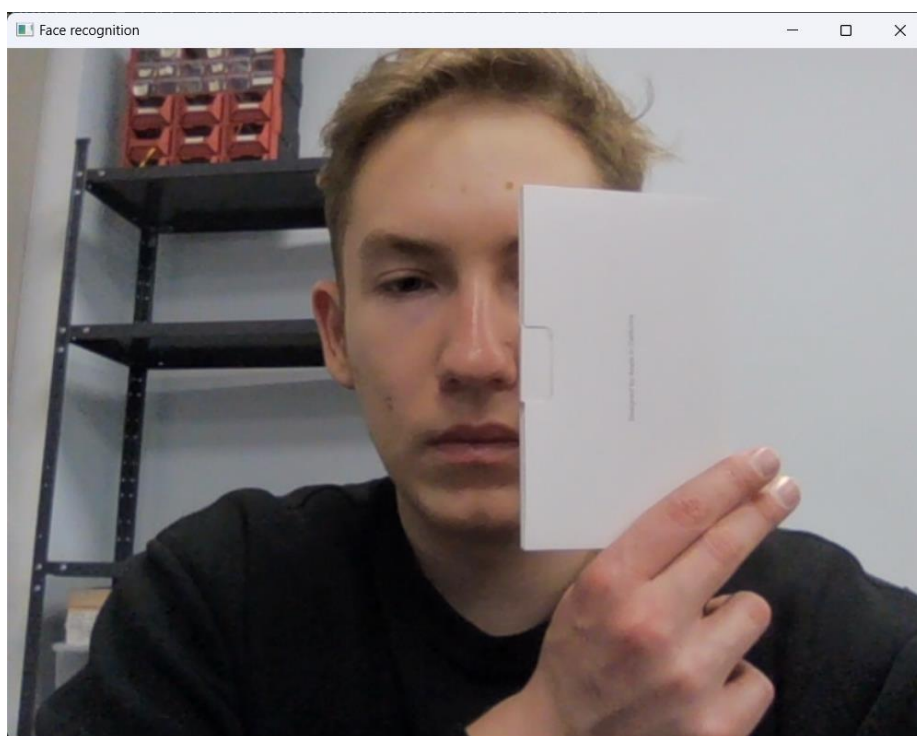


Рисунок 4.10 – Відпрацювання при закритті обличчя > 30%

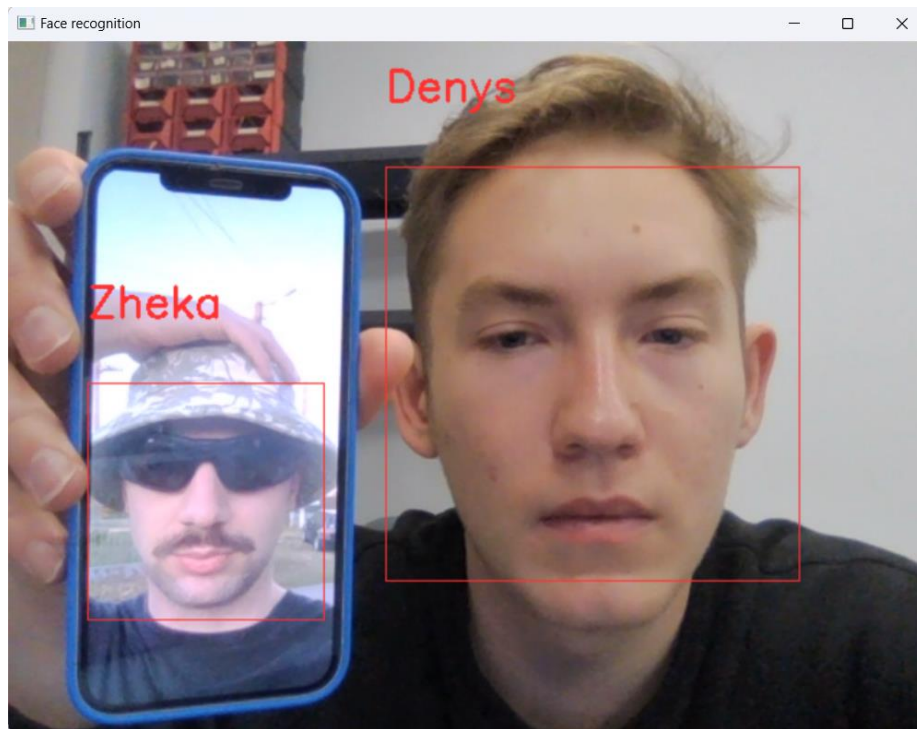


Рисунок 4.11 – Ідентифікація користувача з капелюхом і сонячними окулярами

Також було проведено тестування системи розпізнавання обличчя у умовах поганого освітлення. Швидкість і точність відпрацювання знизились, але не суттєво. Програма коректно змогла розпізнати особистість людини. (рис. 4.12, 4-13)

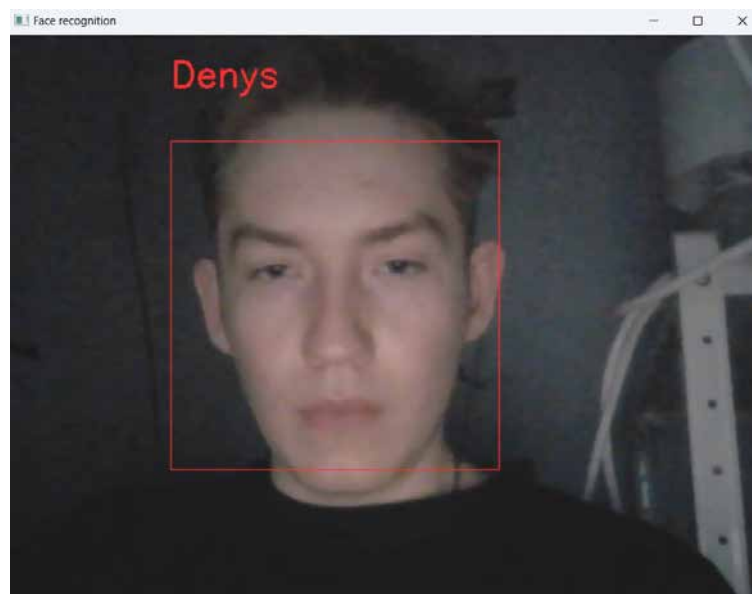


Рисунок 4.12 – Ідентифікація користувача в умовах поганої освітленості

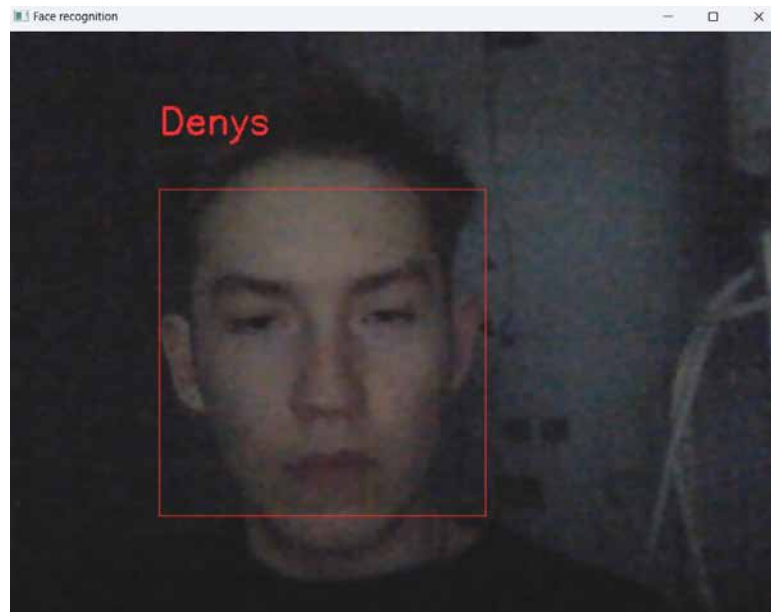


Рисунок 4.13 – Ідентифікація користувача в недостатнього освітлення

При тренуванні моделі було виявлено певну залежність від кількості даних. Спочатку було обрано 100 фотографій для тренування мережі. При малому обсязі фото добитися високої точності розпізнавання досить складно, тому кількість фотографій було збільшено до 500. Завдяки цьому ми змогли отримати значне покращення точності ідентифікації людини. При кількості в 1000 знімків система показала себе досить надійно, забезпечуючи точність близько 92%. Подальше збільшення обсягу даних для тренування не сильно впливало на кінцевий результат, тому оптимальною кількістю, на мою думку, є 1000 фото. Такий набір даних дозволяє системі працювати доволі точно, не потребуючи при цьому значних ресурсів комп'ютера та довгого часу тренування моделі. (рис. 4.14)

## Точність

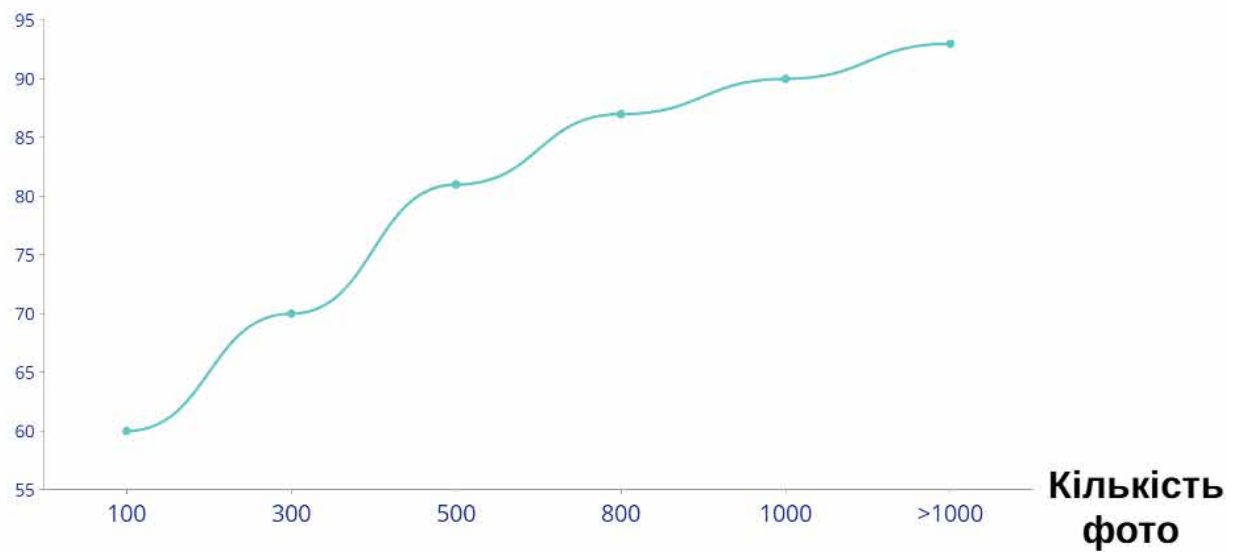


Рисунок 4.14 – Графік залежності точності моделі від кількості даних

### 4.4 Висновки до розділу 4

У ході тестування комп'ютерної системи розпізнавання обличчя з використанням нейронної мережі було детально розглянуто кожен модуль програми. Основний функціонал програми було протестовано в різних умовах використання: при поганому та недостатньому освітленні, з використанням окулярів та маски для обличчя, із частковим закриттям обличчя предметами та за хороших умов.

Результати тестування показали, що точність системи при обробці відеопотоку становить більше 90% правильно розпізнаних кадрів. Також система достойно проявила себе при частковому закритті обличчя і в умовах не дуже хорошо освітлення.

Проте варто відзначити чутливість розробленої системи до сильних немонотонних змін освітленості, а також до змін положення та нахилу розпізнаних

облич, які не враховувалися при захопленні зображень для формування класів облич. Якщо необхідні умови для коректної роботи програми не виконуються, точність класифікації знижується.

Також варто відзначити, що важливу роль також відіграє кількість і якість даних для тренування нейронної мережі. Чим більша і різноманітніша підбірка фото – тим краще.

## **ВИСНОВКИ**

У ході роботи було досліджено методи розпізнавання обличчя людини, їх переваги та недоліки, розглянуто принцип дії нейронних мереж і сфери їх застосування, а також розроблено функціонуючу ком'ютерну систему для

розпізнавання обличчя люди в реальному часі з використанням нейронної мережі.

Програму було реалізовано за допомогою мови програмування Python, середовища програмування PyCharm і бібліотек комп'ютерно зору OpenCV.

У процесі тестування комп'ютерної системи було детально розглянуто кожен модуль програми. Основний функціонал програми було протестовано в різних умовах використання: при поганому та недостатньому освітленні, з використанням окулярів та маски для обличчя, із частковим закриттям обличчя предметами та за хороших умов. Результати тестування показали хорошу точність системи, а саме більше 90% правильно розпізнаних кадрів. Також система достойно проявила себе при частковому закритті обличчя і в умовах не дуже хорошо освітлення. Системи виявилась чутливою до різких змін освітлення та перекриття обличчя більше ніж на 30%.

Окрем було проведено аналіз залежності кількості використаних даних для навчання та кінцевої точності розпізнавання системи. З цього аналізу можна зрозуміти висновок, що для системи потрібен великий об'єм фото з різних ракурсів та за різного освітлення щоб досягти максимальної точності розпізнавання облич.

Система, розроблена в даній роботі, володіє потенціалом для вирішення різноманітних завдань відеоаналітики, зокрема її пряме застосування можливе в системах контролю доступу та ідентифікації особистості. Основними напрямками подальшого розвитку розробленої методики є покращення роботи детектора облич та більша натренованість моделі. Крім того архітектура розробленого додатку забезпечує зручну можливість заміни окремих модулів, що створює більше можливостей для подальшого розширення та вдосконалення системи на різних апаратних платформах.

## **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 511–518. URL: <https://merl.com/publications/TR2004-043>

2. Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29. URL: <https://www.scirp.org/reference/referencespapers?referenceid=1782795>
3. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71–86. URL: <https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>
4. Turk, M., & Pentland, A. (1991). Face recognition using eigenfaces. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–591. URL: <https://ieeexplore.ieee.org/document/139758>
5. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
6. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105. URL: <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45bPaper.pdf>
7. Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 711–720. URL: <https://doi.org/10.1109/34.598228>
8. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297. URL: <https://doi.org/10.1007/BF00994018>
9. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. URL: <https://doi.org/10.1006/jcss.1997.1504>
10. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media. URL: <https://www.oreilly.com/library/view/learning-opencv-3/9781491937990/>
11. Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models.

- IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6), 681–685.  
URL: <https://doi.org/10.1109/34.927467>
12. Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), 2037–2041. URL: <https://doi.org/10.1109/TPAMI.2006.244>
13. Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-Verlag New York, Inc. URL: <https://www.springer.com/gp/book/9780387954424>
14. Gross, R., & Brajovic, V. (2003). An image preprocessing algorithm for illumination invariant face recognition. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 10–15. URL: <https://doi.org/10.1109/AFGR.2003.1216907>
15. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286. URL: <https://doi.org/10.1109/5.18626>
16. Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), 103–108. URL: <https://doi.org/10.1109/34.41386>
17. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. URL: <https://www.springer.com/gp/book/9780387310732>
18. Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press. URL: <http://neuralnetworksanddeeplearning.com/>
19. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. URL: <https://doi.org/10.1109/CVPR.2015.7298682>
20. Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*,

- 14(1), 4–20. URL: <https://doi.org/10.1109/TCSVT.2003.818349>
21. The design of face recognition and tracking for human-robot interaction  
URL:  
[https://www.researchgate.net/publication/323057610\\_The\\_design\\_of\\_face\\_recognition\\_and\\_tracking\\_for\\_human-robot\\_interaction#pf2](https://www.researchgate.net/publication/323057610_The_design_of_face_recognition_and_tracking_for_human-robot_interaction#pf2)
22. Face Recognition (Eigenfaces, Fisherfaces, LBPH) URL:  
<https://medium.com/@seymurqribov05/face-recognition-eigenfaces-fisherfaces-lbph-0b39d41bd54c>
23. Larry Hardesty | MIT News Office (April 14, 2017) . Explained Neural networks  
URL: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
24. Sadaf Saleem (May 15, 2023). Neural Networks in 10mins. Simply Explained!  
URL: <https://medium.com/@sadafsaleem5815/neural-networks-in-10mins-simply-explained-9ec2ad9ea815>
25. Комісаренко Д. С. ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У КОМП'ЮТЕРНИХ СИСТЕМАХ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ . XV МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ МОЛОДИХ ВЧЕНИХ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА» URL:  
<http://econference.nubip.edu.ua/index.php/itete/XV/schedConf/presentations>