

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА


Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

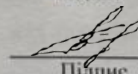
на тему «Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі»

КвРКШ. 303204.23.03.41 ПЗ

Виконав: студент 2 курсу, група КІ2М-23-3

 Богдан МАРТИНЮК  
Підпис Ім'я, прізвище

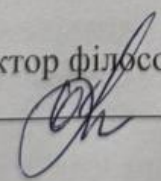
Керівник канд. техн. наук, доцент  
Науковий ступінь, вчене звання

 Володимир ГРИГА  
Підпис Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КІС, доктор філософії, доцент

Ольга ПАВЛОВА

22 05 2025 р. 

Хмельницький, 2025

**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 01 ” 09 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

Богдану МАРТИНЮКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі

Керівник проекту (роботи) Володимир ГРИГА, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Концепція побудови комп'ютерної системи ідентифікації особи на основі нейронних мереж та аналіз існуючих рішень

Базова модель нейромережевої системи для ідентифікації особи

Розробка комп'ютерної системи ідентифікації особи на основі нейронної мережі

Оцінка ефективності комп'ютерної системи ідентифікації особи на основі нейронної мережі

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

## 6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

## КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку та узгодження теми	01.09.2024	виконано
2	Ознайомлення з темою, формулювання мети та задач	01.10.2024	виконано
3	Розділ 1 – аналіз існуючих рішень, постановка задачі	01.11.2024	виконано
4	Розділ 2 – розробка моделі системи	01.12.2024	виконано
5	Написання наукової статті	01.02.2025	виконано
6	Розділ 3 – розробка програмної частини	15.02.2025	виконано
7	Розділ 4 – експерименти та тестування	01.04.2025	виконано
8	Оформлення пояснювальної записки	18.04.2025	виконано
9	Попередній захист	29.04.2025	виконано
10	Захист на засіданні ЕК	До 20.05.2025	

Студент

  
 Підпис

 Богдан МАРТИНЮК  
 Ім'я, прізвище

Керівник роботи

  
 Підпис

 Володимир ГРИГА  
 Ім'я, прізвище

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі.

Автор роботи: Богдан МАРТИНЮК

Керівник роботи: Володимир ГРИГА

Пояснювальна записка: 90 с., 52 рис., 5 табл., 7 дод., 83 джерела.

ПЕРЕЛІК КЛЮЧОВИХ СЛІВ: НЕЙРОННА МЕРЕЖА, ІДЕНТИФІКАЦІЯ ОСОБИ, ОПТИМІЗАЦІЯ, КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ, СИСТЕМА БЕЗПЕКИ.

Об'єктом дослідження є процеси ідентифікації особи на основі нейронних мереж у системах безпеки, зокрема в спеціалізованих комп'ютерних системах для забезпечення доступу до захищених ресурсів.

Предметом дослідження є система ідентифікації особи на основі нейронної мережі для забезпечення безпеки в автоматизованих системах доступу.

Метою кваліфікаційної роботи магістра є підвищення ефективності та точності ідентифікації особи в системах безпеки за допомогою використання нейронних мереж для розпізнавання особливих біометричних характеристик, таких як обличчя, відбитки пальців або райдужка ока.

Для розв'язання поставлених задач використовувалися математичні та аналітичні методи дослідження, комп'ютерне моделювання, методи оцінки ефективності, а також методи оцінки точності та швидкості системи ідентифікації.

Наукова новизна отриманих результатів: набула подальшого розвитку базова модель системи ідентифікації особи на основі нейронної мережі, яка включає використання сучасних архітектур глибоких нейронних мереж для покращення точності та швидкості ідентифікації. Це дозволило розробити нові методи оптимізації навчання нейронної мережі для зменшення помилок ідентифікації в реальних умовах.

Набув подальшого розвитку метод інтеграції нейронних мереж в мультиагентні системи для підвищення ефективності ідентифікації особи, що дозволяє значно зменшити час реагування на запити та підвищити рівень безпеки.

Практична цінність роботи полягає в тому, що запропонована система ідентифікації особи на базі нейронної мережі може бути інтегрована в існуючі системи контролю доступу в різних галузях, таких як фінансові установи, державні органи або в систему Інтернету речей. Це дозволить підвищити ефективність доступу до захищених ресурсів та зменшити ризик несанкціонованого доступу до важливих даних.

У першому розділі виконано огляд існуючих методів і технологій ідентифікації особи, зокрема через нейронні мережі. Проаналізовано переваги та недоліки традиційних методів ідентифікації (паролі, картки) порівняно з методами біометричної ідентифікації. Окремо досліджено принципи роботи нейронних мереж у контексті обробки біометричних даних.

У другому розділі було розроблено абстрактну модель для системи ідентифікації особи, яка включає використання різних типів нейронних мереж (конволюційних, рекурентних) для обробки зображень та інших біометричних даних. Модель також враховує такі фактори, як освітлення, кут огляду та інші параметри, що можуть вплинути на якість ідентифікації.

У третьому розділі були визначені основні критерії для оптимізації роботи системи ідентифікації особи. Для цього була розроблена математична модель, що включає вагові коефіцієнти для різних факторів, таких як точність, швидкість і надійність. Запропоновано архітектуру системи, яка включає використання нейронних мереж для розпізнавання образів і інтеграцію цих мереж в мультиагентну систему для забезпечення високої ефективності ідентифікації.

У четвертому розділі була проведена оцінка ефективності розробленої системи ідентифікації. Проведено порівняння результатів роботи системи, заснованої на базовій моделі і нейронній мережі, з іншими відомими методами ідентифікації. Було виявлено, що запропонована система забезпечує більш високу точність і швидкість ідентифікації, а також менше піддається впливу факторів навколишнього середовища, таких як зміни освітлення чи кути огляду.

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....</b>	<b>5</b>
<b>ВСТУП .....</b>	<b>6</b>
<b>1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ІДЕНТИФІКАЦІЇ ОСОБИ .....</b>	<b>8</b>
1.1 Основні аспекти ідентифікації особи за зображенням .....	8
1.2 Етапи процесу розпізнавання обличчя.....	9
1.3 Програмні засоби розпізнавання обличчя.....	13
1.4 Висновки.....	27
<b>2 ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ЛЮДСЬКИХ ОБРАЗІВ .....</b>	<b>28</b>
2.1 Нейронні мережі як інструмент машинного навчання.....	28
2.2 Архітектура системи автоматичного розпізнавання обличчя.....	46
2.3 Висновки.....	46
<b>3 РОЗРОБКА МОДЕЛІ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ .....</b>	<b>52</b>
3.1 Огляд сучасних згорткових мереж для розпізнавання .....	52
3.2 Внесення змін до архітектури системи ідентифікації особи.....	65
3.3 Висновки.....	68
<b>4 РЕАЛІЗАЦІЯ ОПТИМІЗОВАНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ОСОБИ .....</b>	<b>69</b>
4.1 Підготовка середовища для розробки .....	69
4.2 Підготовка робочого середовища для розробки .....	74
4.3 Алгоритми оцінювання продуктивності нейронної мережі.....	78
4.4 Створення класу для процесу навчання та тестування .....	81
4.5 Програмна реалізація та тренування системи розпізнавання обличчя .....	84

	4
4.6 Аналіз ефективності розробленої системи .....	86
4.7 Висновки.....	93
<b>ВИСНОВКИ</b> .....	94
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	96
<b>ДОДАТОК А</b> Копія тези доповіді на міжнародній конференції.....	102
<b>ДОДАТОК Б</b> Презентація до захисту кваліфікаційної роботи .....	102
<b>ДОДАТОК В</b> Виділення признаков.....	1133
<b>ДОДАТОК Г</b> Модифікація мережі .....	1154
<b>ДОДАТОК Д</b> Розробка та навчання моделей Г.....	11615
<b>ДОДАТОК Е</b> Тестування моделей.....	116
<b>ДОДАТОК Ж</b> Розробка та навчання моделей Г.....	117

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- ЗВО – Заклад вищої освіти
- РБ – Розумний будинок
- ІТ – Інформаційні технології
- ІО – Інтернет об'єктів
- НМ – Нейронні мережі
- ЛС – Логічна система
- ОТ – Оптимізація технологій
- ПЗ – Програмне забезпечення
- ТП – Технічні параметри
- АІ – Штучний інтелект
- ВК – Вхідний канал
- ВТ – Вихідний тест
- ВП – Вхідні параметри
- ТМ – Технічне моделювання
- ММ – Математичне моделювання
- КТ – Комп'ютерне тестування
- ІМО – Інтелектуальна модель обробки
- ПН – Параметри нейронної мережі
- БМ – Базова модель
- ПС – Програмна система
- ДР – Дослідницька робота

## ВСТУП

Сучасні технології ідентифікації особи відіграють ключову роль в забезпеченні безпеки та зручності у різних сферах, таких як фінансові послуги, доступ до особистих даних, а також в системах контролю доступу. Одним із найперспективніших підходів до вирішення задачі ідентифікації є використання нейронних мереж, оскільки вони здатні обробляти великий обсяг даних та виявляти складні закономірності, що робить процес ідентифікації більш точним і надійним [1].

Актуальність цієї роботи обумовлена зростаючою потребою в удосконаленні систем ідентифікації особи, що відповідають високим вимогам безпеки та ефективності. З традиційними методами, такими як паролі чи картки доступу, все частіше виникають проблеми з безпекою, оскільки вони можуть бути вкрадені, забуті або підроблені. Нейронні мережі, у свою чергу, дозволяють значно підвищити точність ідентифікації завдяки здатності до навчання на великих наборах даних та їх здатності адаптуватися до змінних умов.

У рамках цієї роботи пропонується розробка спеціалізованої комп'ютерної системи ідентифікації особи на базі нейронної мережі, яка використовує новітні методи машинного навчання для підвищення точності та швидкості ідентифікації. Така система забезпечить більш високий рівень захисту, адже вона здатна враховувати різноманітні параметри, що ускладнюють підробку даних або несанкціонований доступ.

Метою роботи є розробка ефективної системи ідентифікації особи на основі нейронних мереж, яка дозволить забезпечити високий рівень безпеки і точності при ідентифікації осіб у різних сферах. Це досягається через використання алгоритмів глибокого навчання для аналізу різноманітних ознак, таких як зображення обличчя, голос, дактилоскопічні дані або інші біометричні показники [2].

Об'єктом дослідження є процеси ідентифікації особи за допомогою нейронних мереж, що використовують біометричні дані для забезпечення точності та безпеки.

Предметом дослідження є розробка спеціалізованої комп'ютерної системи ідентифікації особи на базі нейронної мережі, що забезпечить інтеграцію різних біометричних методів для більш надійного результату.

Методи дослідження. У роботі застосовано наступні методи:

- математичні методи для побудови моделей нейронних мереж;
- аналітичні методи для аналізу результатів тестування і ефективності запропонованої системи;
- комп'ютерне моделювання для створення та налаштування нейронної мережі;
- методи оцінки точності та швидкості ідентифікації в реальних умовах.

Наукова новизна роботи полягає в подальшому розвитку технологій ідентифікації на основі нейронних мереж шляхом інтеграції кількох біометричних даних та використання новітніх методів глибокого навчання для підвищення точності та надійності системи.

Практична цінність роботи полягає в тому, що запропонована система ідентифікації особи на базі нейронних мереж може бути використана для створення безпечних та зручних систем доступу до різних сервісів, включаючи банківські та фінансові платформи, а також для покращення рівня безпеки у сфері державного управління та корпоративних мережах.

Рекомендації з використання результатів роботи. Результати дослідження можуть бути застосовані для розробки нових систем ідентифікації, зокрема в сфері контролю доступу, криміналістики, безпеки в Інтернеті, а також для покращення ефективності існуючих біометричних систем.

Публікації. За темою роботи було опубліковано кілька статей у фахових наукових виданнях, що висвітлюють досягнуті результати та перспективи розвитку біометричних технологій.

Структура та об'єм роботи. Робота складається з вступу, чотирьох розділів, висновків та додатків. Загальний обсяг роботи – 96 сторінок, включаючи 52 рисунки, 5 таблиць та 83 джерела літератури.

# 1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ІДЕНТИФІКАЦІЇ ОСОБИ

## 1.1 Основні аспекти ідентифікації особи за зображенням

Розпізнавання людських образів на зображеннях є однією з найважливіших задач у сфері комп'ютерного зору, яка стала поштовхом для розвитку теорії розпізнавання образів. Хоча протягом останніх десятиліть проводилося безліч наукових досліджень, спрямованих на вдосконалення методів і алгоритмів у цій галузі, універсального підходу, який би гарантував точне та стабільне розпізнавання людських образів за будь-яких умов, досі не створено.

Процес розпізнавання обличчя включає два основні етапи:

1. Детекція обличчя – автоматичне виявлення області обличчя на зображенні або у відеопотоці.
2. Ідентифікація особи – визначення, кому саме належить знайдене обличчя (у разі використання бази зареєстрованих осіб).

Виявлені обличчя можуть використовуватися для подальшого аналізу, зокрема для створення 3D-моделей, відстеження міміки, аналізу виразів обличчя, визначення вікових характеристик і статі, а також для реалізації механізмів аутентифікації та верифікації особи [3].

Застосування технологій розпізнавання обличчя має велике значення у створенні людино-комп'ютерного інтерфейсу, автоматизації систем безпеки та контролю доступу до важливих об'єктів. Оскільки в реальних умовах важливо швидко ідентифікувати наявність обличчя у відеопотоці, актуальною задачею є розробка високопродуктивних алгоритмів аналізу відеоданих у реальному часі.

У межах даної кваліфікаційної роботи буде реалізовано програмну систему, яка аналізуватиме відеопотік, що надходить із веб-камери, і на основі отриманих даних формуватиме одну з наступних відповідей:

- відеопотік не містить людських образів;
- відеопотоці виявлено обличчя, яке не вдалося розпізнати;

– у відеопотоці присутнє розпізнане обличчя, яке відповідає записам у базі даних.

Розроблена система працюватиме в реальному часі, забезпечуючи автоматичне виявлення та розпізнавання осіб. Для цього буде використано попередньо навчену нейронну мережу, яка пройде навчання на спеціально підготовленому наборі зображень. Це дозволить досягти високої точності роботи алгоритму навіть у складних умовах, наприклад, при зміні освітлення або кута повороту обличчя.

Окрім основної функції розпізнавання, передбачено можливість інтеграції з системами безпеки, що дозволить застосовувати технологію для обмеження доступу до приміщень або верифікації користувачів у різних інформаційних системах. У ході реалізації буде досліджено сучасні алгоритми машинного навчання, такі як методи згорткових нейронних мереж (CNN), а також підходи до оптимізації обробки відеопотоку для мінімізації затримок і підвищення продуктивності [4].

Завдяки цьому розробка матиме практичну цінність як для автоматизованих систем контролю, так і для інтеграції в смарт-технології майбутнього.

## 1.2 Етапи процесу розпізнавання обличчя

Існує кілька основних підходів до вирішення задачі детектування людських образів, серед яких можна виокремити найбільш поширені методи.

Традиційний метод базується на аналізі ключових точок та характерних рис обличчя. Алгоритми цього типу визначають розташування, форму та розміри окремих елементів, таких як очі, ніс, вилиці та щелепа. Отримані параметри використовуються для пошуку інших зображень, що мають схожі характеристики.

Альтернативні методи передбачають нормалізацію галереї зображень людських образів, після чого здійснюється стиснення інформації, зберігаючи лише ті дані, які є важливими для процесу розпізнавання. Надалі тестове зображення порівнюється з отриманими характеристиками поверхні. Одним із перших

успішних методів був підхід, заснований на зіставленні шаблонів, який дозволяв створювати узагальнений опис обличчя шляхом обробки його поверхневих особливостей.

Алгоритми розпізнавання поділяються на два ключові підходи. Геометричний підхід базується на аналізі конкретних структурних характеристик обличчя, тоді як фотометричний підхід, який має статистичну природу, перетворює зображення у числовий набір ознак, що дозволяє здійснювати порівняння з наявними шаблонами для мінімізації відмінностей.

Також методи розпізнавання людських образів можна класифікувати на дві великі групи: цілісні та засновані на ознаках. Цілісні методи орієнтуються на ідентифікацію всього обличчя як єдиного об'єкта, тоді як підходи, що використовують аналіз окремих ознак, розбивають обличчя на складові елементи, такі як очі, ніс та рот, і оцінюють їх не лише окремо, а й у контексті їхнього просторового розташування відносно інших частин обличчя.

Популярні методи розпізнавання людських образів включають кілька підходів, таких як аналіз головних компонент із застосуванням методу власних граней, лінійний дискримінантний аналіз, еластичне зіставлення через алгоритм риб'ячого обличчя, приховану марковську модель, полілінійне підпросторове навчання, тензорне представлення та динамічне зіставлення за допомогою нейронних мереж.

Тривимірне розпізнавання обличчя передбачає використання 3D-сенсорів для збору інформації про форму поверхні обличчя. Ця інформація потім використовується для виявлення характерних особливостей обличчя, таких як форма повік, носа та підборіддя. Однією з переваг цього методу є те, що він не так чутливий до змін освітленості, як інші методи, а також здатний розпізнавати обличчя під різними кутами, включаючи профіль. Використання тривимірних точок на обличчі значно підвищує точність розпізнавання, оскільки дозволяє враховувати більш детальну інформацію. Розвиток цієї технології був можливий завдяки появі вдосконалених датчиків, що краще за інших справляються із

захопленням 3D-зображень. Такі датчики можуть бути інтегровані в мікросхему CMOS, де кожен датчик займає свою частину спектра.

Однак навіть найсучасніші методи зіставлення 3D-даних можуть бути чутливими до емоційних виразів. Для вирішення цієї проблеми дослідники з Technion використали інструменти метричної геометрії, щоб видалити вплив емоцій із поверхні обличчя за допомогою ізометрії [5].

Новий підхід включає використання трьох камер, орієнтованих під різними кутами для захоплення тривимірного зображення обличчя. Одна камера спрямована на обличчя прямо, інша - в профіль, а третя – під кутом. Спільно ці камери здатні ефективно відстежувати та розпізнавати обличчя в реальному часі.

Аналіз текстури шкіри – ще один новий напрямок, який використовує деталі шкіри, отримані через цифрові або скановані зображення. Цей метод перетворює природні лінії, візерунки та плями на шкірі в математичний простір, що дозволяє точніше ідентифікувати особу. Аналіз текстури шкіри працює подібно до розпізнавання обличчя, де малюнок шкіри обробляється як відбиток і розбивається на дрібніші ділянки. Потім за допомогою алгоритмів цей відбиток перетворюється в математичні вимірювання, що дозволяє системі виявляти будь-які лінії, пори та текстуру шкіри, чого неможливо досягти тільки за допомогою традиційних методів розпізнавання людських образів.

Дослідження показали, що додатковий аналіз текстури шкіри може суттєво підвищити ефективність систем розпізнавання людських образів, збільшуючи точність на 20-25%. Одним з найбільш популярних методів для виявлення людських образів на зображеннях є метод Віоли-Джонса (Viola-Jones), який здобув визнання завдяки своїй високій швидкості і ефективності. Цей підхід дозволив реалізувати розпізнавання людських образів в реальному часі, що знайшло застосування в таких практичних додатках, як цифрові камери і програмне забезпечення для керування фотоальбомами.

Метод Віоли-Джонса здатний виявляти обличчя, що знаходяться на передньому плані, не розмиті, добре освітлені і знаходяться в нормальній орієнтації. Він працює на зображеннях з фіксованою роздільною здатністю і

дозволяє коригувати зображення для підготовки до процесу виявлення. Це включає нормалізацію, коригування яскравості, а також пересування вікна по зображенню для отримання обмежувальних рамок, що містять обличчя [6]. Однак для обробки обличчя під різними кутами розроблені кілька різних категорій для кожного кута огляду. Для розв'язання проблеми з фрагментами обличчя, що можуть бути частково приховані або розмиті, застосовуються методи зменшення повторення, як-от немаксимальне придушення.

Основні принципи роботи алгоритму Віоли-Джонса включають інтегроване представлення зображень на основі властивостей Хаара, адаптивне підвищення (AdaBoost) для побудови класифікаторів, а також каскадне об'єднання класів для пришвидшення процесу виявлення людських образів. Ці особливості забезпечують високу швидкість і точність розпізнавання, що робить метод надзвичайно ефективним для використання в реальному часі.

Переваги методу включають високу швидкість виявлення, точність понад 90% для фронтальних зображень та спостереження об'єктів під невеликими кутами (до  $30^\circ$ ), а також низьку ймовірність помилкового розпізнавання. Однак метод має й недоліки, серед яких тривалий процес навчання, зменшення точності при великих кутах нахилу голови, а також чутливість до освітлення.

Метод власних людських образів (Eigenfaces) використовує аналіз головних компонент (PCA) для зменшення розмірності даних без значної втрати інформації. Простір власних людських образів створюється шляхом застосування методу PCA до набору навчальних зображень, що дозволяє проектувати нові зображення на цей простір і знаходити найближчі до них зображення з навчальної вибірки. Основною перевагою цього методу є висока точність при дотриманні ідеальних умов, а також можливість зберігання і пошуку зображень у великих базах даних. Проте метод потребує великої обчислювальної потужності для обчислення власних векторів і вимог до освітлення та попередньої обробки зображень.

Лінійний дискримінантний аналіз (LDA) або дискримінант Фішера використовує проекцію простору зображень на новий простір, що дозволяє групувати схожі зображення ближче одне до одного. Цей метод значно покращує

кластеризацію і дозволяє досягти високої точності розпізнавання навіть у змінних умовах освітлення. Алгоритм стійкий до змін освітлення і може досягати рівня точності до 99%.

Методи, що ґрунтуються на штучних нейронних мережах, використовуються для автоматичного вибору ознак і взаємозалежностей між ними. Серед таких мереж можна виділити багатошарові перцептрони, мережі з радіальними базисними функціями та згорточні нейронні мережі (CNN) [7], які є ефективними для задач розпізнавання людських образів. Проте якість розпізнавання знижується із збільшенням кількості класів для класифікації.

Методи сегментації кольору шкіри також застосовуються для локалізації людських образів, виділяючи ділянки зображення, що мають колір, подібний до шкіри людини. Для цього створюються набори правил, які допомагають відрізнити пікселі шкіри від інших елементів зображення.

У рамках цієї роботи ми виберемо методи, засновані на штучних нейронних мережах, оскільки вони демонструють високу ефективність та здатні адаптуватися до різних умов і змін у зображеннях.

### 1.3 Програмні засоби розпізнавання обличчя

Під час проведення аналізу були протестовані наступні основні функціональні блоки:

- виявлення людських образів на фотографіях (Face Detection);
- порівняння людських образів (Face Comparison);
- визначення атрибутів обличчя (стать, вік, етнічна приналежність, емоції);
- пошук схожих людських образів – здатність системи розпізнати обличчя певної людини серед групи зображень, тобто знайти найбільш схоже на задане обличчя.

Кожен з перерахованих блоків тестувався за кількома критеріями, що є важливими для оцінки ефективності системи:

- точність, що вимірюється як відсоток співпадіння між обличчями;
- швидкість роботи системи – час, який проходить від моменту відправки запиту до отримання результату;
- вартість виконання кожної функції розпізнавання;
- додаткові можливості і функціонал, пов'язані з аналізом обличчя.

Azure Face API [8] – це потужна хмарна служба для розпізнавання людських образів, яка є частиною набору інтелектуальних сервісів Azure Cognitive Services від Microsoft. Цей інструмент використовує передові алгоритми, які дозволяють виявляти, перевіряти, ідентифікувати і аналізувати обличчя на зображеннях і відео.

Коли служба Azure Face API виявляє обличчя на зображенні, вона визначає координати прямокутної області, в якій воно знаходиться. Крім того, якщо це необхідно, система може також визначити кілька характеристик, пов'язаних із зображеними людьми, таких як положення голови, стать, вік, емоції, наявність волосся на обличчі, чи є на особі окуляри, а також визначити ймовірність того, що це обличчя належить конкретній людині з бази даних.

Як приклад, система може визначити, що на зображенні знаходиться усміхнений чоловік 36 років, і його основною емоцією є щастя [9]. Цей підхід дозволяє точно аналізувати різні атрибути обличчя, надаючи можливість застосовувати технологію в широкому спектрі задач. Дане дослідження продемонстровано на рисунку 1.1.

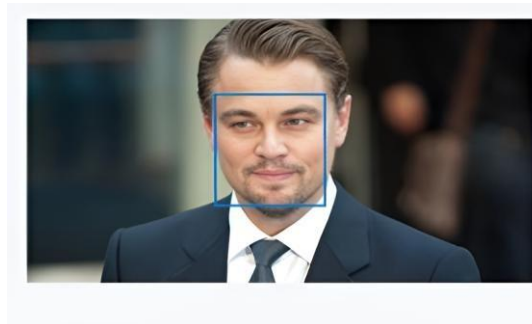


Рисунок 1.1 – Виявленні обличчя на фото [9]

Face ID є технологією розпізнавання людських образів, яка здатна визначати їхню наявність на зображенні, аналізувати їхнє положення та точно окреслювати межі у вигляді рамок. Окрім базового розпізнавання, система також ідентифікує ключові точки обличчя, зокрема підборіддя, очі, брови, кути рота та форму носа.

Додатково Face ID може аналізувати різноманітні характеристики особи, серед яких стать, приблизний вік, ступінь виразності посмішки, нахил і поворот голови. Крім того, система здатна визначати ступінь розмитості зображення та оцінювати стан очей: чи вони відкриті або закриті, а також чи присутні на обличчі окуляри.

Усі отримані дані подаються у вигляді структурованого звіту у форматі JSON, що дозволяє легко інтегрувати інформацію в різні програмні додатки та автоматизовані системи. Це забезпечує зручність обробки результатів і використання їх у подальших етапах аналізу, таких як ідентифікація особи, оцінка емоційного стану чи персоналізація користувацького досвіду (рис.1.2).

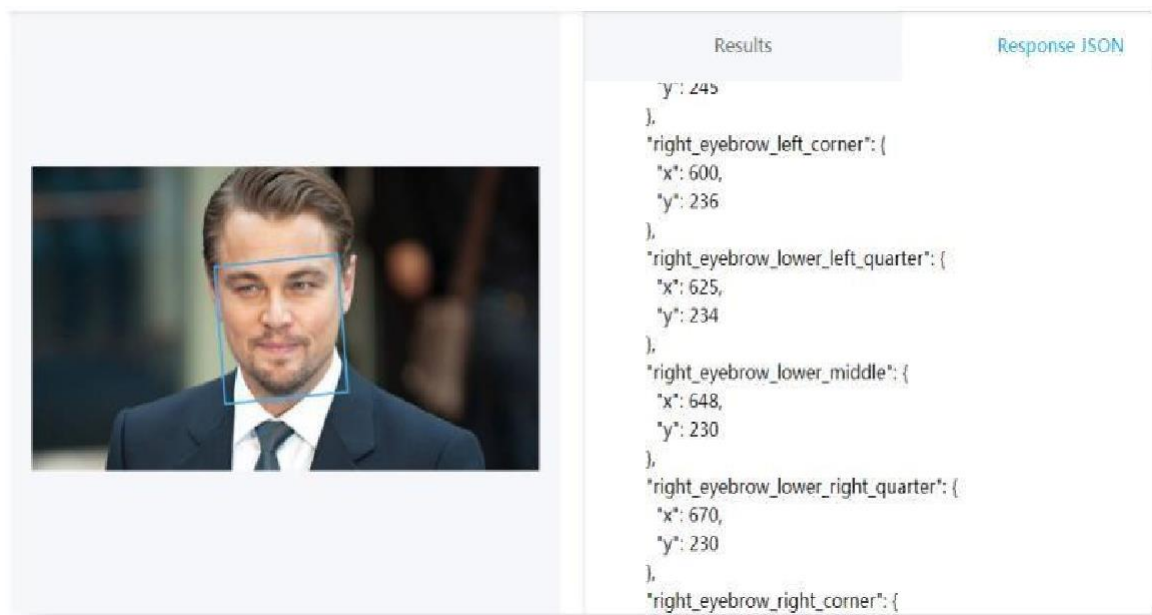


Рисунок 1.2 – Визначення координат та атрибутів [9]

Для технології Face ID середній час, необхідний для визначення наявності людського обличчя на зображенні при обробці великої вибірки фото, становить

приблизно 1,4 секунди [10]. Це забезпечує швидке та ефективне розпізнавання, що є важливим фактором для систем, які працюють у реальному часі .

В Azure Face для порівняння зображень використовується функція «перевірка обличчя». Вона аналізує, чи належать два обличчя на фото одній і тій самій людині, і визначає межі кожного з них, повертаючи відповідні координати. Наприклад, у випадку, коли на зображенні присутні різні особи, система коректно ідентифікує їх як окремих людей [11]. Час обробки запиту становить приблизно 394 мілісекунди. Порівняння продемонстровано на рисунку 1.3.

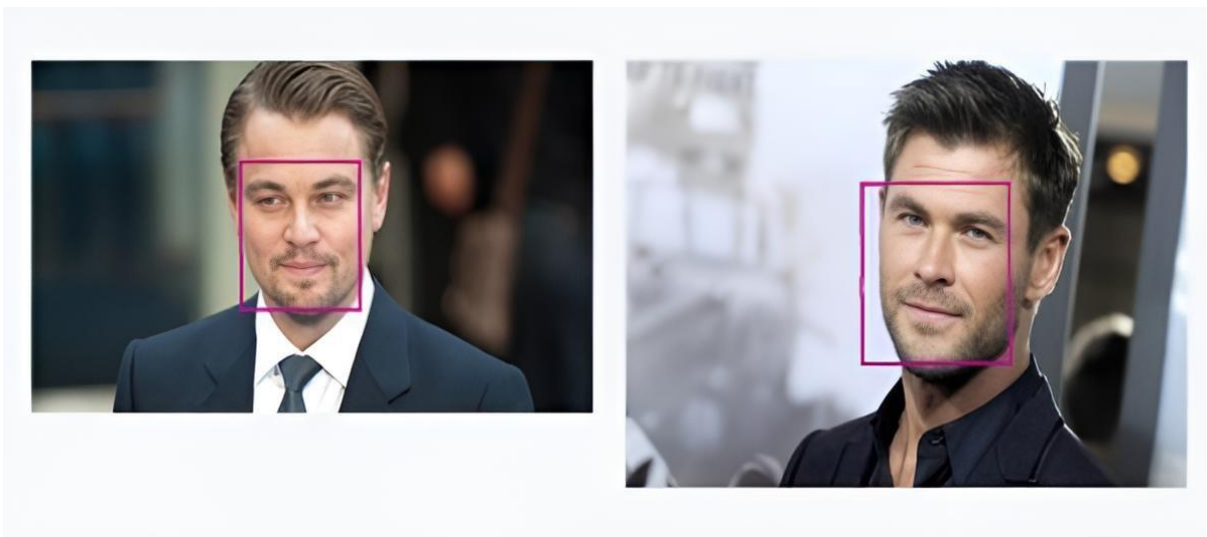


Рисунок 1.3 – Порівняння людських образів [11]

FaceID здатний порівнювати два зображення для визначення, чи належать вони одній і тій самій людині. Користувач може завантажити фото безпосередньо з пристрою або вказати URL-адресу, щоб перевірити зображення на схожість. Система виконує аналіз людських образів, визначаючи відсотковий рівень подібності (наприклад, якщо результат становить 59,2%, то ймовірно, що це різні особи). Час обробки запиту зазвичай складає 454 мілісекунди.

Крім того, Azure Face API має вбудовану функцію розпізнавання емоцій, що дозволяє ідентифікувати широкий спектр виразів обличчя, зокрема нейтральний стан, радість, гнів, презирство, огиду, смуток, страх і здивування. Цей механізм працює на етапі аналізу, до виконання основних дій. Під час обробки система

визначає ключові атрибути обличчя, обводячи його прямокутною рамкою, та надає відповідні коефіцієнти впевненості для кожної з виявлених емоцій. Такий підхід дозволяє застосовувати технологію в різних сценаріях, зокрема у сфері безпеки, аналітики та користувацьких інтерфейсів, що реагують на емоційний стан людини. На рисунках 1.4 та 1.5 продемонстровано порівняння подібностей та оцінка достовірності по кожній емоції.

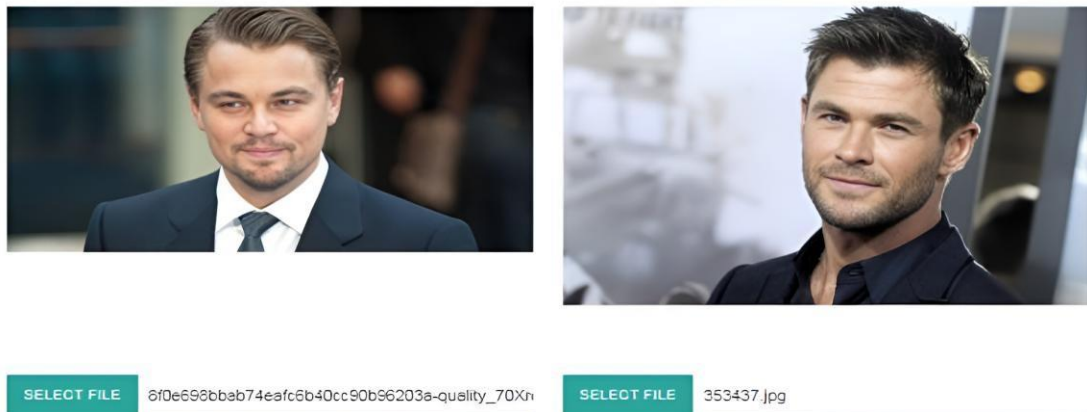


Рисунок 1.4 – Порівняння подібності людських образів [11]

Функція розпізнавання емоцій наразі знаходиться в процесі розробки, оскільки вона не входить до стандартного пакета послуг з ідентифікації людських образів.

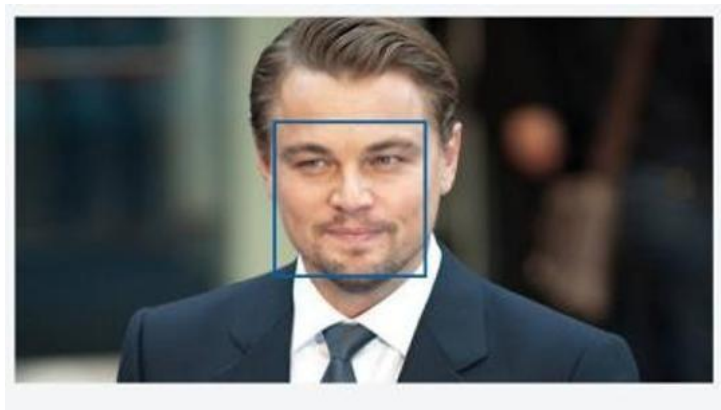


Рисунок 1.5 – Оцінка достовірності для кожної емоції [12]

Проте, за потреби, можна створити індивідуальне рішення, яке здатне визначати такі емоційні стани, як радість, нейтральний вираз обличчя, здивування, смуток, огиду, гнів та страх. Це може бути корисним у багатьох сферах, зокрема у створенні інтерактивних користувацьких інтерфейсів, маркетингових дослідженнях та безпекових системах.

Окрім цього, сервіс Azure Face API має можливість порівнювати цільове обличчя із набором доступних для пошуку зображень, щоб ідентифікувати максимально схожі обличчя. В системі передбачено два режими роботи:

- режим `matchPerson` – цей режим використовується для пошуку схожих людських образів серед фотографій, при цьому система застосовує фільтрацію, щоб повертати результати лише для конкретної особи;
- режим `matchFace` – у цьому режимі не застосовується фільтр за особистістю, тобто система виводить список виявлених людських образів, незалежно від того, чи належать вони одній і тій самій людині

На даний момент функціонал пошуку однакових людських образів не інтегрований у базовий пакет Face ID [12]. Проте, якщо така можливість необхідна для реалізації певного проекту, її можна впровадити як додаткову функцію. Це розширить можливості застосування системи в таких сферах, як біометрична безпека, системи контролю доступу та аналіз поведінки користувачів у різних галузях.



Рисунок 1.6 – Пошук подібних людських образів [11]

Для виконання порівняння між цілим зображенням і набором фотографій, після обробки API пошуку аналізує отримані дані та надає список зображень, які мають найбільшу схожість із цільовим фото. До кожного знайденого обличчя система додає відповідні коефіцієнти подібності, що дозволяють оцінити ступінь відповідності між зображеннями [13].

У процесі роботи алгоритм враховує різні параметри, такі як пропорції обличчя, положення рис, освітлення та кут нахилу голови, що допомагає підвищити точність визначення схожості. Таким чином, користувач отримує не лише список схожих зображень, а й детальну інформацію про рівень подібності, що може бути використано для подальшого аналізу або автоматизованих рішень у системах розпізнавання.

Результати порівняння виводяться у вигляді структурованих даних, що можуть бути інтегровані в різні додатки або бази даних для подальшої обробки та прийняття рішень. Це може бути корисним у сферах безпеки, маркетингу, контролю доступу та інших напрямках, де необхідна висока точність ідентифікації осіб. Це продемонстровано на рисунку 1.7 та 1.8 (результати запису в JSON файл.

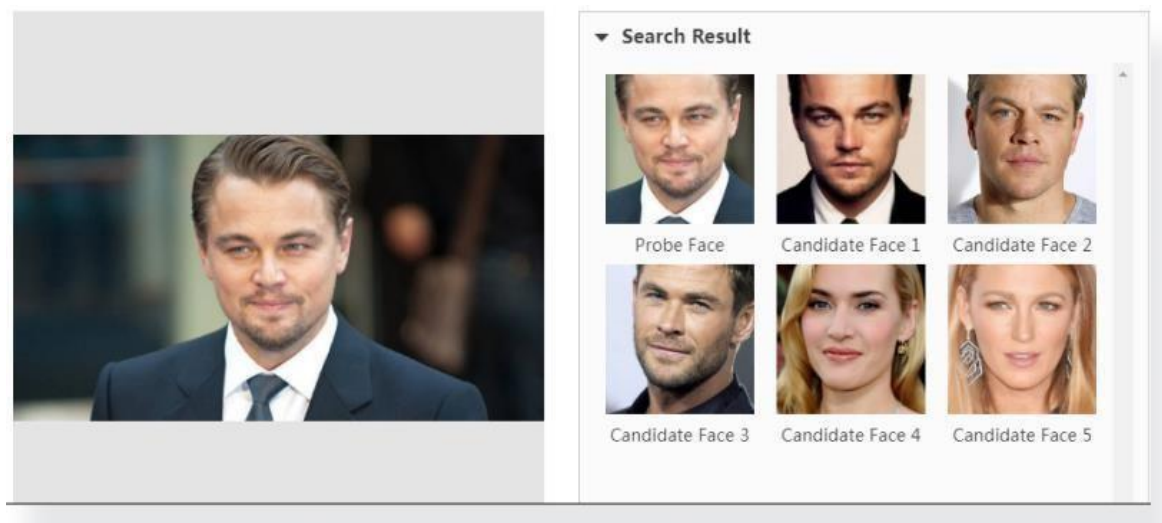


Рисунок 1.7 – Порівняння з групою кандидатів [11]

У цьому прикладі можна побачити, що система успішно ідентифікувала одну й ту саму людину на двох різних знімках. Фото 1 було використано як запит для пошуку, і серед усіх проаналізованих зображень найбільшу ймовірність

відповідності отримало фото 2. Для інших кандидатів система визначила низький рівень збігу, що свідчить про їхню невідповідність цільовій особі.

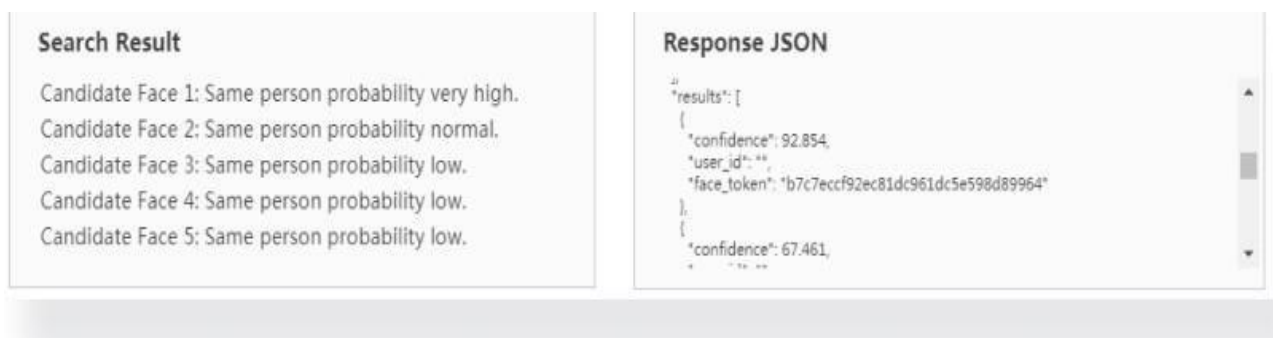


Рисунок 1.8 – Результати пошуку і відповідь системи в форматі JSON [13]

У цьому прикладі можна побачити, що система успішно ідентифікувала одну й ту саму людину на двох різних знімках. Фото 1 було використано як запит для пошуку, і серед усіх проаналізованих зображень найбільшу ймовірність відповідності отримало фото 2. Для інших кандидатів система визначила низький рівень збігу, що свідчить про їхню невідповідність цільовій особі.

Cloud Vision API – це потужний хмарний сервіс від Google, який використовує попередньо навчені моделі для аналізу зображень. Цей інструмент здатний виконувати широкий спектр завдань: від розпізнавання об'єктів та людських образів до ідентифікації тексту на фото, автоматичної генерації метаданих та інших функцій аналізу зображень. У рамках цього дослідження ми зосередимося на можливостях, що стосуються виявлення та порівняння людських образів.

Функціонал розпізнавання людських образів у Google Cloud Vision API дозволяє знаходити на фото одне або декілька людських образів. Крім цього, система визначає загальні характеристики зображення, наприклад, ступінь освітленості, рівень розмиття та можливі спотворення, що можуть вплинути на якість аналізу. Однак варто зазначити, що цей сервіс не підтримує ідентифікацію особистості за обличчям, а лише здійснює базове виявлення наявності людських образів на зображенні.

Cloud Vision API працює за принципом загального розпізнавання об'єктів, у тому числі й людських образів. При аналізі знімка система повертає набір даних, який включає текстовий опис виявленого об'єкта, рівень впевненості у правильності розпізнавання та координати прямокутної області, у якій знаходиться обличчя. Вся інформація представлена у структурованому форматі JSON, що робить її зручною для подальшого використання у програмних застосунках, системах безпеки, автоматизованих системах контролю та інших сферах [14].

Завдяки хмарній природі сервісу Cloud Vision API користувачі можуть отримати доступ до потужних алгоритмів машинного навчання без необхідності налаштовувати власні моделі чи витратити ресурси на їхнє навчання. Це робить систему привабливою для компаній, які потребують ефективних інструментів аналізу зображень для своїх бізнес-процесів. На рисунку 1.9 та 1.10 продемонстровано розпізнавання та вивчення наявності обличчя на фото за допомогою Google Cloud Vision API.

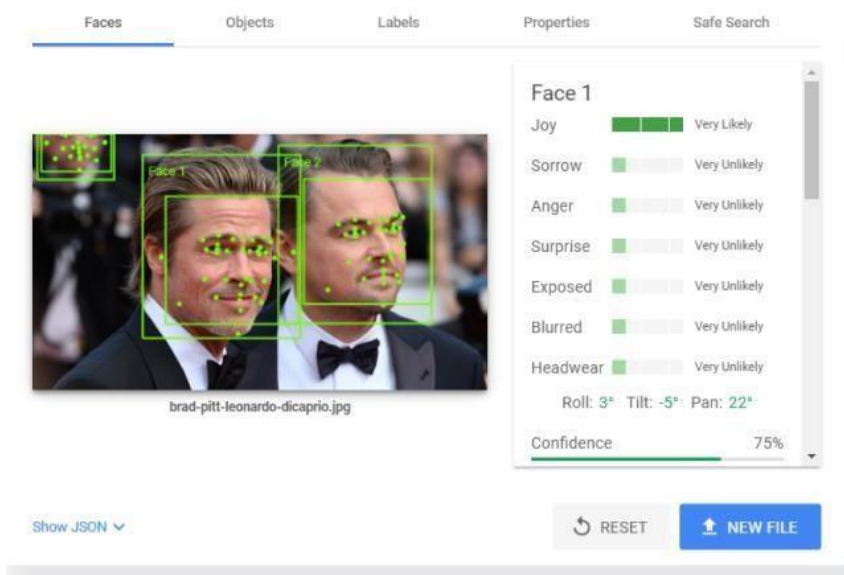


Рисунок 1.9 – Розпізнавання людських образів в Google Cloud Vision API [14]

На відміну від Face ID, Cloud Vision API не підтримує функції перевірки та порівняння людських образів, а лише визначає їхню присутність на фото. Ця система може розпізнавати, скільки людей зображено на знімку, але не проводить детального аналізу їхньої схожості [15]. Таким чином, Cloud Vision API більше

орієнтований на загальне розпізнавання об'єктів і не може бути використаний для автентифікації осіб або перевірки тотожності між знімками.

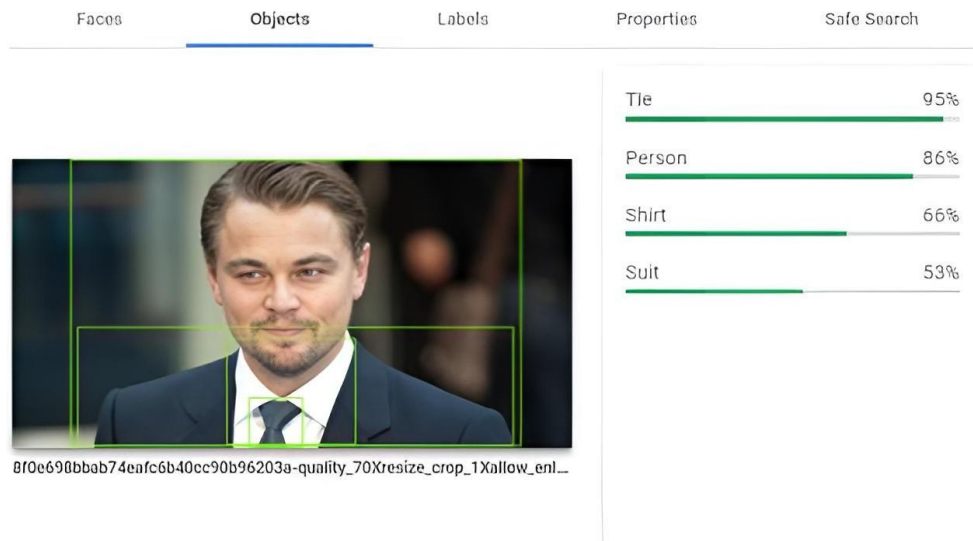


Рисунок 1.10 – Визначення наявності людських образів на зображенні [15]

Якщо розглядати функцію порівняння, Face ID дозволяє зіставляти два зображення, визначаючи, чи належать вони одній і тій самій особі. Для цього система аналізує обличчя на обох фото та обчислює рівень їхньої подібності. Якщо відсоток збігу менший за 80%, алгоритм робить висновок, що на знімках різні люди. Якщо ж показник високий (наприклад, 93,1%, як у наведеному прикладі), система підтверджує, що на двох зображеннях з великою ймовірністю представлена та сама людина.

Окрім точності, важливим параметром є швидкість обробки запитів. У наведеному тестовому випадку система Face ID проаналізувала два зображення та видала результат протягом 1,163 секунди. Це доводить, що алгоритм працює достатньо швидко, що є критично важливим для застосування в реальному часі, наприклад, у системах безпеки, мобільних додатках або автоматизованих сервісах контролю доступу.

Таким чином, Face ID забезпечує комплексне розпізнавання людських образів із можливістю їхньої ідентифікації, тоді як Cloud Vision API більше підходить для базового аналізу наявності людських образів на зображенні без глибокої перевірки їхньої тотожності (рис.1.11).

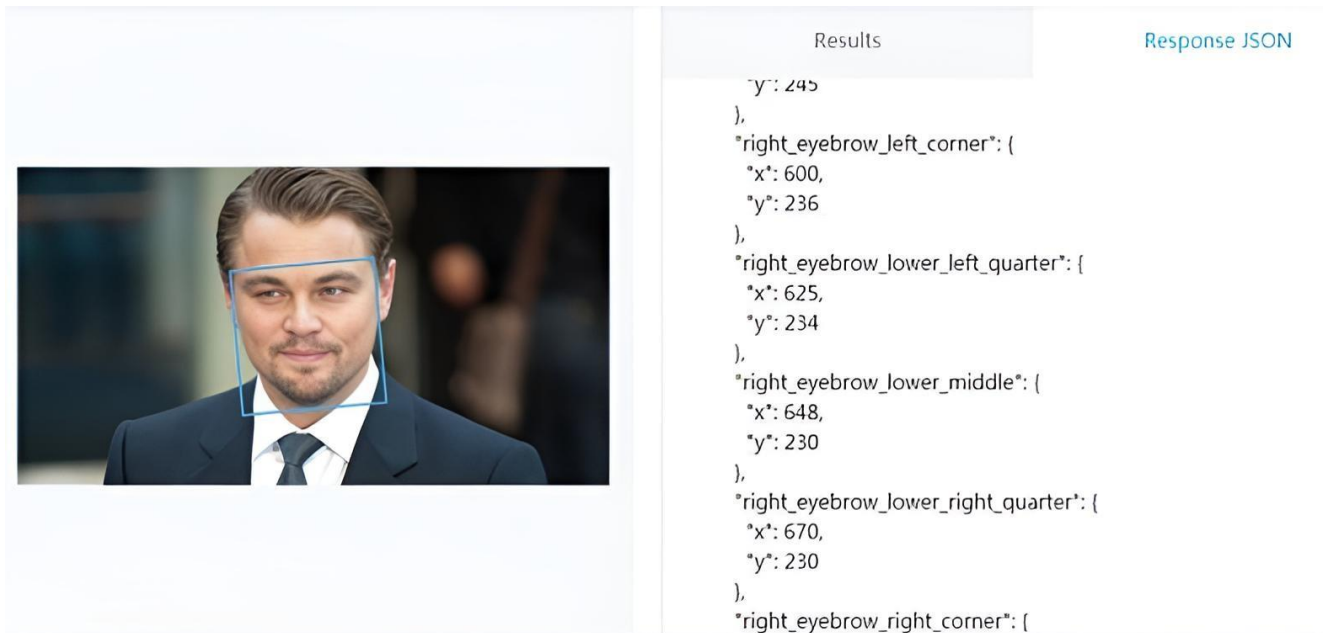


Рисунок 1.11 – Визначення координат і атрибутів в Cloud Vision API [15]

Попри значну зовнішню схожість акторів, система визначила, що рівень їхньої подібності становить 57,6%, що означає, що їхні обличчя не є ідентичними. При цьому швидкість обробки запиту склала 0,459 секунди, що свідчить про ефективність алгоритму навіть у випадках складного аналізу схожості.

Cloud Vision API застосовує алгоритми комп'ютерного зору для детального аналізу людських образів на фотографіях. Система не лише розпізнає присутність людей у кадрі, але й окреслює їхні обличчя прямокутниками, точно визначаючи координати ключових точок, таких як очі, вуха, ніс, рот та інші структурні елементи. Це дозволяє ідентифікувати розташування основних рис обличчя та використовувати ці дані для подальшого аналізу.

Окрім цього, Cloud Vision API здатний оцінювати емоційний стан людини на зображенні, застосовуючи імовірнісні бали для визначення емоцій. Наприклад, система може визначити рівень вираженості щастя, смутку, злості чи здивування на обличчі людини, що дає змогу використовувати ці дані у різних сферах, таких як маркетинг, аналіз настрою користувачів або створення інтерактивних додатків [16].

Додатково Cloud Vision API аналізує загальні характеристики знімка, зокрема:

- рівень експозиції – чи є зображення затемненим або, навпаки, перенасиченим світлом;
- ступінь розмитості – визначає, наскільки чітким є зображення, що важливо для точності подальшого аналізу;
- наявність масок або перешкод – система може виявляти, чи приховані обличчя елементами, такими як маски, окуляри чи інші предмети.

Cloud Vision API забезпечує не лише базове розпізнавання людських образів, а й комплексний аналіз емоційного стану, якості зображення та ключових особливостей, що робить його потужним інструментом для роботи з цифровими фотографіями. На рисунку 1.12-1.14 наглядно показано фото оцінки різних/однакових людей та оцінку їх емоцій [17].

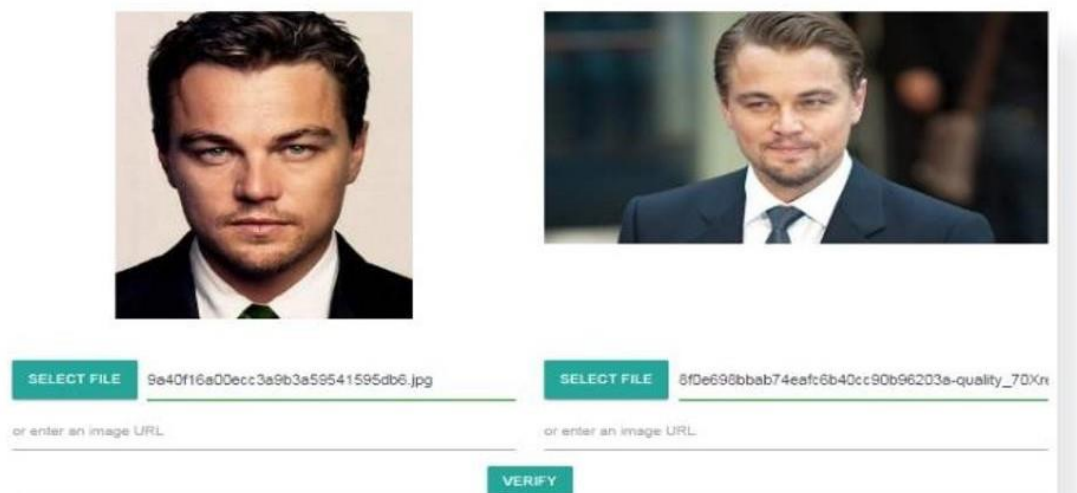


Рисунок 1.12 – Фото однієї людини [16]

Серед програмних рішень для розпізнавання людських образів варто виділити KLIK – додаток, розроблений для операційної системи iOS, а також EmguCVRecognizer, який працює на основі бібліотеки EmguCV.

Програма EmguCVRecognizer є одним із аналогів, що вирішує задачу ідентифікації осіб. Вона розроблена на мові C#, що, порівняно з рішеннями на C++, може призводити до меншої швидкодії при виконанні складних обчислень. Проте основним аспектом, який потребує вдосконалення, є рівень достовірності

розпізнавання – на даний момент він становить близько 80%, що може бути недостатнім для високоточних сценаріїв використання.

Що стосується програми KLIK, яка працює під управлінням iOS, вона пропонує зручний інтерфейс та інтегровані алгоритми для обробки людських образів на фотографіях. На рисунку 1.14 представлено приклад роботи цієї програми, що демонструє процес розпізнавання людських образів на зображеннях [18].

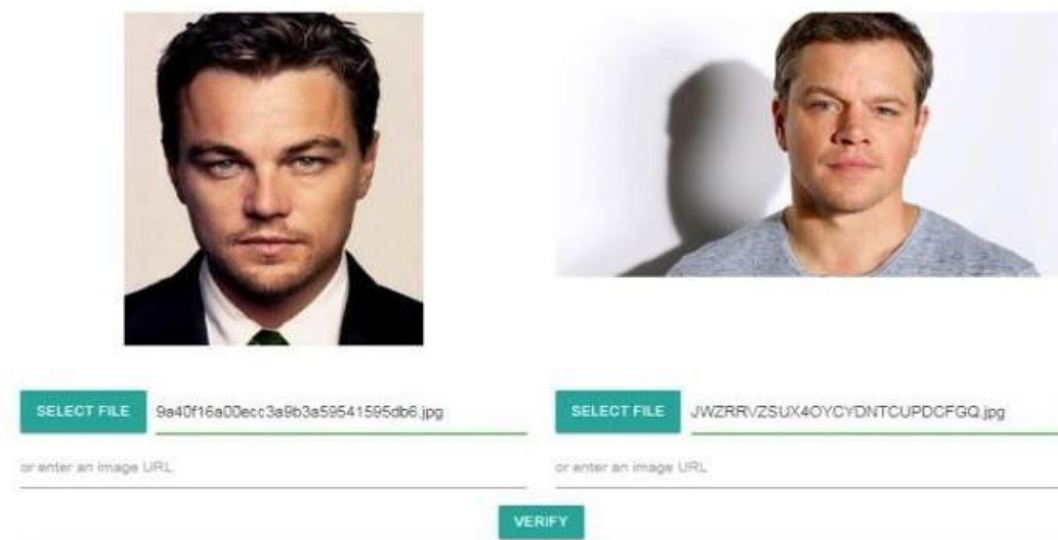


Рисунок 1.13 – Фото двох різних людей [18]

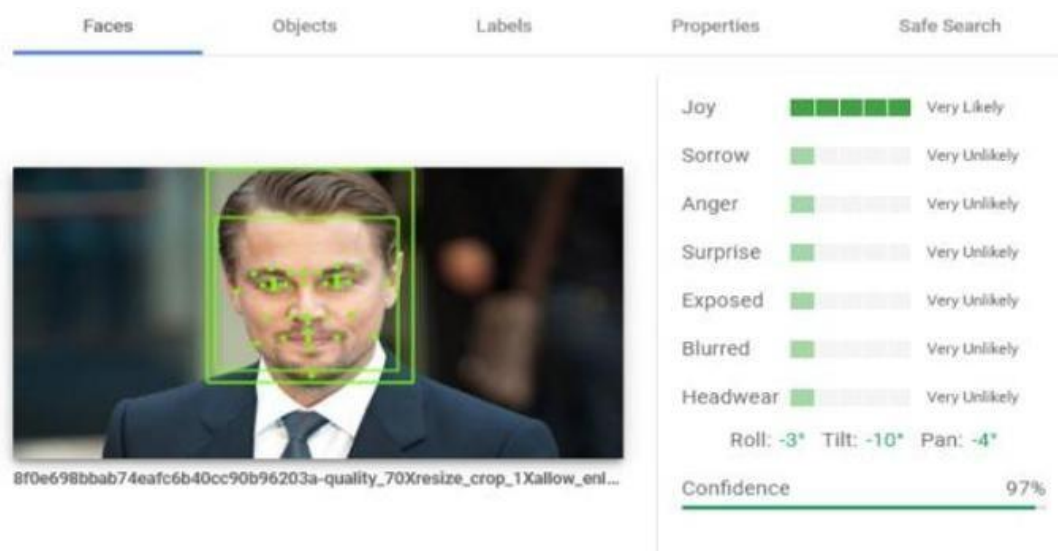


Рисунок 1.14 – Імовірнісні оцінки для емоцій [18]

Таким чином, обидві програми мають свої переваги та недоліки. KLIK зосереджений на мобільному використанні з інтеграцією в iOS-середовище, тоді як EmguCVRecognizer надає можливості для розпізнавання в .NET-додатках, хоча й поступається за продуктивністю та точністю розпізнавання. Скріншот KLIK зображено на рис. 1.16.

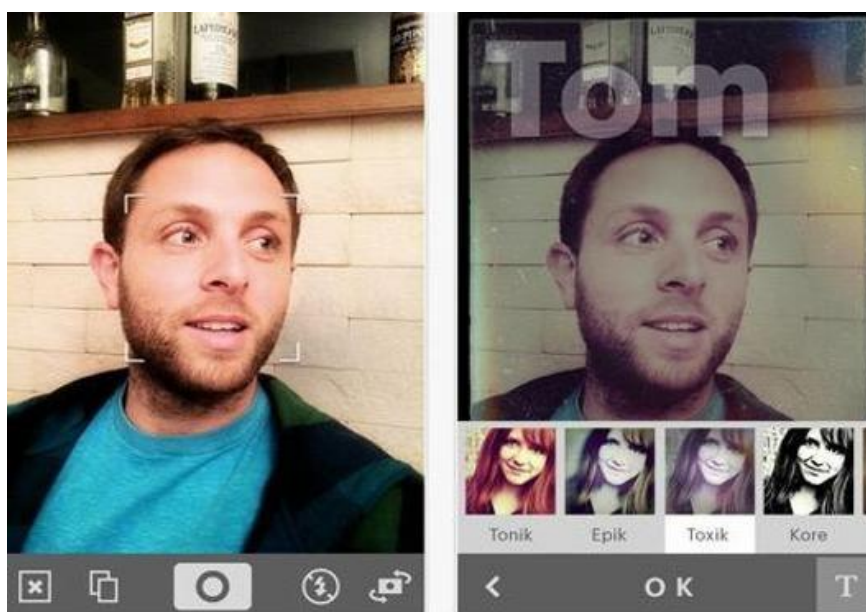


Рисунок 1.15 – Скріншот програми KLIK [18]

Компанія Face.com є одним із лідерів у сфері розробки технологій розпізнавання людських образів. Саме ця компанія постачає алгоритми ідентифікації для таких великих платформ, як Facebook. Протягом кількох місяців розробники працювали над спеціальним проектом (його справжня назва не розголошується) та представили інноваційне рішення у вигляді мобільного додатка KLIK, який призначений для пристроїв під керуванням iOS 4.3+.

Основна перевага KLIK – це розпізнавання людських образів у режимі реального часу. Програма аналізує осіб, що потрапляють у поле зору камери смартфона, і на основі попередньо збережених зображень визначає їхню особу. Якщо людина була позначена раніше або входить до списку друзів користувача у Facebook, система здатна її ідентифікувати автоматично.

Продукт вражає швидкістю роботи: у багатьох випадках розпізнавання відбувається настільки оперативно, що не потрібно навіть натискати кнопку спуску затвора, аби зафіксувати особу. Проте, якщо зображення є складним для аналізу (наприклад, на ньому кілька людей або воно має низьку якість), обробка може займати до однієї хвилини.

На момент запуску KLIK перебував на експериментальному етапі розробки (версія 0.9). Незважаючи на це, він демонстрував ефективну роботу навіть у складних умовах, таких як погане освітлення. За словами розробників, середня точність розпізнавання людських образів у системі Face.com становить 91,3%, що є дуже високим показником у порівнянні з багатьма іншими рішеннями у сфері біометричної ідентифікації [19].

Таким чином, KLIK представляє потужний мобільний інструмент, який дозволяє користувачам швидко ідентифікувати людей у кадрі, інтегруючись із соціальними мережами та забезпечуючи комфортний і зручний досвід використання.

#### 1.4 Висновки

У даному розділі виконано комплексний аналіз сучасних методів і технологій розпізнавання обличчя, особливо в контексті використання нейронних мереж. Виявлено переваги біометричних систем порівняно з традиційними методами автентифікації, зокрема вищу точність, зручність та стійкість до підробок. Розглянуто існуючі API та сервіси (Azure Face, Face ID, Cloud Vision), а також алгоритми (Viola-Jones, Eigenfaces, CNN), що демонструють ефективність розпізнавання в реальному часі. Встановлено, що найперспективнішим напрямом є застосування глибокого навчання та конволюційних нейронних мереж.

## 2 ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ЛЮДСЬКИХ ОБРАЗІВ

Розпізнавання людських образів є окремим випадком загальної проблеми ідентифікації образів. Його можна описати таким чином: у систему надходить певний об'єкт  $x$ , який необхідно ідентифікувати. У контексті завдання біометричного розпізнавання цей об'єкт представлений у вигляді цифрового зображення обличчя [20-24].

Припустимо, що  $x$  може належати одному з класів, які входять до множини  $A = \{a_1, a_2, \dots, a_n\}$ , де кожен клас пов'язаний із конкретною особою, наприклад, її ім'ям або унікальним ідентифікатором. Умовою є неперетинання цих класів, тобто кожне обличчя належить лише одному класу.

Функція  $f(x)$ , яка відповідає за процес розпізнавання, аналізує вхідне зображення та повертає ідентифікатор класу  $A$ , до якого належить особа, а також рівень достовірності отриманого результату. Якщо ж система не знаходить відповідності, вона має визначити, що дане зображення не співпадає з жодним відомим класом [25].

### 2.1 Нейронні мережі як інструмент машинного навчання

Концепція нейронних мереж спочатку була задумана як математична модель, що імітує принцип роботи людського мозку. Відомо, що мозок людини складається з нейронів, які безперервно обмінюються інформацією за допомогою електричних сигналів, передаючи їх через спеціальні з'єднання – синапси. При цьому не всі нейрони пов'язані між собою, а міцність їхнього зв'язку може змінюватися.

У процесі навчання людини відбувається формування нових зв'язків між нейронами або посилення вже існуючих. Чим міцніше ці зв'язки, тим легше інформація передається між нейронами, що забезпечує автоматизацію навичок. Наприклад, зв'язки між нейронами, що відповідають за базові вміння, набуті в ранньому дитинстві, такі як ходьба або мовлення, є дуже міцними. Протягом життя

зв'язки можуть посилюватися або слабшати залежно від того, наскільки активно використовується певна навичка.

Ця ідея стала основою для створення перших штучних нейронних мереж, зокрема перцептрона (рис.2.1), який є однією з базових моделей глибокого навчання. Перцептрон також відомий як повнозв'язна нейронна мережа, оскільки його структура передбачає, що кожен нейрон одного шару пов'язаний із кожним нейроном наступного шару.

Пізніше, на основі досліджень роботи зорової кори людини, була розроблена модель згорткової нейронної мережі (ЗНМ), яка спеціалізується на обробці та розпізнаванні зображень. Такі мережі часто містять у своїй структурі повнозв'язні шари, проте їхньою головною особливістю є застосування згорткових операцій, що дозволяють виділяти ключові особливості об'єктів на зображеннях [26].

Перцептрон складається з кількох послідовних шарів, кожен з яких включає штучні нейрони. Між нейронами існують зв'язки, які мають вагові коефіцієнти, що визначають силу впливу одного нейрона на інший. Чим вищий коефіцієнт ваги, тим сильніший взаємозв'язок між нейронами, що безпосередньо впливає на точність і ефективність навчання нейронної мережі.



Рисунок 2.1 – Загальний вигляд структури перцептрону [26]

Повнозв'язну нейронну мережу можна розглядати як своєрідний «чорний ящик», який отримує вхідні дані, перетворює їх за допомогою прихованих шарів і видає результат на виході. У цій структурі перший шар називається вхідним, останній – вихідним, а всі проміжні – прихованими шарами.

Вхідні нейрони виконують найпростішу функцію – вони лише приймають певні числові значення ( $X$ ) і передають їх далі вглиб мережі. Однак у наступних шарах починається складна обробка даних, яка дозволяє мережі виконувати нелінійні перетворення і приймати рішення.

Кожен нейрон прихованого шару складається з двох ключових компонентів: суматора та функції активації. Суматор відповідає за отримання сигналів від усіх нейронів попереднього шару, кожен із яких має свою вагу, що визначає його вплив на кінцевий результат (рис.2.2). Вхідні значення перемножуються на відповідні вагові коефіцієнти, після чого підсумовуються [27].

Щоб зробити модель більш гнучкою, у рівняння вводиться порогове значення, також відоме як зміщення (*bias*). Його можна уявити як додатковий нейрон із фіксованим значенням  $+1$ , який дозволяє налаштовувати поведінку моделі та виконувати афінні перетворення. Це забезпечує адаптивність нейронної мережі, дозволяючи їй точніше налаштовувати кореляцію між вхідними та вихідними даними.

Функція активації, яка є наступним етапом обробки, відіграє важливу роль у навчанні нейронної мережі. Вона визначає, чи повинен нейрон передавати сигнал далі та в якій формі. Існує кілька типів функцій активації, кожна з яких має свої особливості та використовується залежно від конкретного завдання:

- сигмоїдна функція (Sigmoid) – добре підходить для задач, де необхідно отримати ймовірність належності до певного класу;
- ReLU (Rectified Linear Unit) – найбільш поширена функція активації, яка дозволяє нейромережі ефективно навчатися, нівелюючи проблему затухаючого градієнта;
- гіперболічний тангенс (Tanh) – подібний до сигмоїдної функції, але працює краще, оскільки його значення змінюються в ширшому діапазоні  $(-1;1)$ .

Завдяки поетапній обробці інформації в повнозв'язній нейронній мережі відбувається перетворення вхідних даних та формування відповідей, що дозволяє

використовувати такі моделі для розпізнавання образів, обробки мовлення, прогнозування даних та інших складних аналітичних завдань.

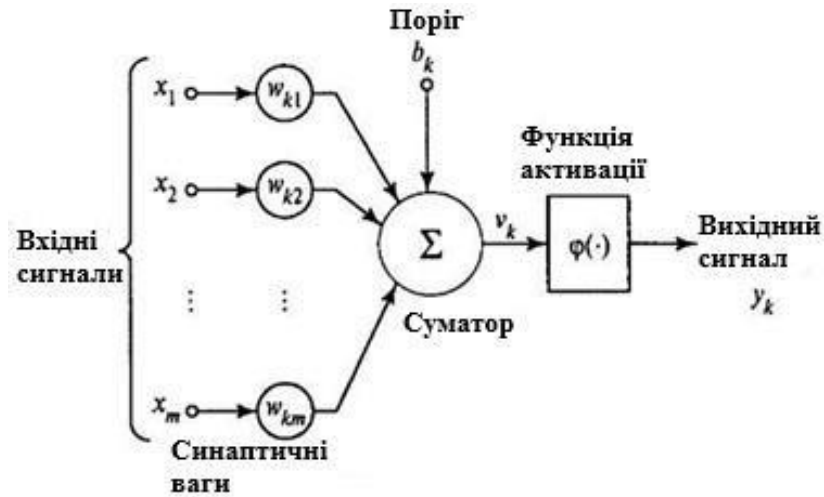


Рисунок 2.2 – Формальна модель нейрону [27]

Функція активації виконує важливу роль у нормалізації вихідного сигналу, згладжуючи значення, отримані на етапі підсумовування. Вона обмежує діапазон вихідних значень, що дозволяє мережі працювати стабільно та уникати надмірних коливань у переданих даних.

У більшості випадків значення на виходах нейронів знаходяться в межах  $[0,1]$  або  $[-1,1]$ , що залежить від конкретної функції активації, яка використовується в моделі. Це дозволяє мережі ефективно обробляти інформацію, адаптувати ваги та коригувати свої рішення під час навчання.

Математично вихідне значення нейрона розраховується за спеціальною формулою, яка визначає спосіб обчислення нового сигналу на основі отриманих входів.

Математично вихідне значення нейрона розраховується за спеціальною формулою (2.1), яка визначає спосіб обчислення нового сигналу на основі отриманих входів.

$$f(w_1x_1 + w_2x_2 + \dots + w_mx_m + b), \quad (2.1)$$

де  $f$  – використовувана функція активації;

$w_{tm}$  – вага між поточним нейроном і нейроном  $m$  попереднього шару;

$x_m$  – значення на нейроні  $m$  попереднього шару;

$b$  – зміщення на поточному нейроні.

У нейронній мережі функція активації може бути єдиною для всієї архітектури або змінюватися для кожного окремого шару. Проте, найчастіше для прихованих шарів обирають одну конкретну функцію, а для вихідного шару підбирають іншу, що залежить від специфіки поставленого завдання [28-30].

Сьогодні існує великий вибір функцій активації, серед яких найбільш популярні: логістична функція (сигмоїда), гіперболічний тангенс, ReLU (Rectified Linear Unit) і PReLU (Parametric ReLU). Вибір конкретної функції залежить від структури нейромережі, типу вхідних даних та мети навчання. Не існує універсального підходу до вибору активаційної функції – зазвичай це визначається шляхом експериментів або використанням найбільш ефективних рішень з попередніх досліджень.

Одна з найпоширеніших функцій – сигмоїда. Вона добре працює у мережах з невеликою кількістю прихованих шарів (зазвичай не більше шести). Основною причиною цього є те, що сигмоїдні функції видають значення, обмежені інтервалом  $(0,1)$ . Через це під час навчання, коли значення передаються через кілька шарів, вони поступово стають дуже малими. У результаті навчання відбувається повільніше, оскільки градієнти в перших шарах практично не змінюються, а основне коригування ваг відбувається лише у останніх шарах. Це явище називають проблемою зниклого градієнта, і саме через нього в глибоких нейромережах сигмоїда використовується рідко.

Гіперболічний тангенс (див. формулу 2.3) є функцією активації, яка має схожі властивості з сигмоїдою, але з деякими відмінностями. Основною особливістю є те, що його значення обмежені в діапазоні від  $-1$  до  $1$ . Це дає можливість отримувати негативні значення на виході, що може бути корисним у певних задачах, де така поведінка є необхідною.

Гіперболічний тангенс часто використовується в ситуаціях, коли мережа повинна моделювати дані з двома ознаками, де як позитивні, так і негативні значення можуть мати важливе значення для правильного навчання згідно формули (2.2). Він дозволяє більш ефективно вирішувати проблему зниклого градієнта в порівнянні з сигмоїдною функцією, оскільки його похідна набагато більша і стійка, що допомагає прискорити навчання в мережах середньої глибини.

$$f(x) = \frac{1}{N} \sum_{i=1}^N \frac{e^{|d_i - y_i|}}{\max(|y_i|, \varepsilon)}, \quad (2.2)$$

де  $d_i$  – еталон;

$y_i$  – прогноз;

$\varepsilon$  – захист від ділення на 0.

Для нейронних мереж з великою кількістю шарів на прихованих шарах часто застосовують функцію активації ReLU (Rectified Linear Unit), оскільки вона виявляється надзвичайно ефективною в таких умовах. Вона дозволяє значно прискорити навчання мережі, а також зменшити ймовірність виникнення проблеми зниклого градієнта, що є поширеною в глибоких мережах. Однак, навіть при невеликій кількості шарів, ReLU часто застосовується замість інших функцій активації, таких як гіперболічний тангенс, завдяки своїм перевагам в обчислювальній ефективності та швидкості навчання.

Раніше в деяких випадках використовували гіперболічний тангенс, однак з часом було виявлено, що ReLU дає більш стабільні та точні результати, що призвело до її широкого застосування в більшості сучасних глибоких нейронних мережах [31-34].

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad (2.3)$$

де градієнт: 0 ( $x < 0$ ) або 1 ( $x \geq 0$ ).

Іноді модифікація ReLU, відома як PReLU (Parametric ReLU), може показувати кращі результати. Відмінність цієї функції активації від стандартної ReLU полягає в додаванні параметра, який навчається разом з іншими параметрами мережі. Цей параметр дозволяє мережі адаптуватися до різних типів даних, що може підвищити її гнучкість і точність в порівнянні зі стандартним ReLU, особливо в складних задачах.

$$f(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.4)$$

де,  $a \in (0,1)$  — малий коефіцієнт «пропуску» (типово 0.01).

У задачах класифікації при непересічних класах використовується функція softmax на вихідному шарі (див. формулу 2.5). Вона переводить значення вихідних нейронів в значення, ближче до імовірнісних, враховуючи значення на всіх вихідних нейронах.

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^m e^{x_j}}, i = 1, \dots, m, \quad (2.5)$$

де  $x \rightarrow$  – вихідний вектор нейронної мережі;

$m$  – кількість значень в вихідному векторі.

У завданнях регресії, тобто в задачах, де вихідне число це і є відповідь і його не потрібно якось інтерпретувати, як, наприклад, в задачах класифікації, на вихідному шарі може використовуватися лінійна функція активації;

$$f(x) = x, \quad (2.6)$$

де  $f(x)$  – лінійна функція активації;

$x$  – вхідне значення, що без змін передається на вихід.

Отже, процес, у якому нейронна мережа проходить через шари і обчислює значення на кожному нейроні, називається процесом прямого поширення або

прямим ходом. На кожному етапі мережа генерує результат, який потім передається на наступний шар, і цей процес триває до останнього шару, де отримується фінальний вихід.

Навчання нейронної мережі являє собою задачу оптимізації, в рамках якої проводиться налаштування параметрів мережі, таких як ваги та зміщення, для досягнення бажаного результату. Це досягається через обчислення помилки за допомогою обраної функції помилки (критерію якості), що дозволяє виміряти відхилення між отриманим результатом і бажаним. Після цього помилки на нейронах останнього шару обчислюються, а через метод градієнтного спуску відбувається оновлення значень ваг зв'язків, що ведуть до цих нейронів [35-38].

Щоб визначити, на скільки саме потрібно змінити кожен вагу, обчислюється похідна функції помилки по цій вазі. Ця похідна вказує на нахил функції помилки, що відображає напрямок, у якому необхідно рухатися, щоб зменшити помилку. І для того, щоб ефективно коригувати ваги, застосовується коефіцієнт швидкості навчання, який визначає, на скільки саме буде змінюватися кожна вага в напрямку до мінімуму функції помилки. Цей коефіцієнт можна розглядати як "крок" на шляху оптимізації параметрів мережі та вираховується за формулою (2.7).

$$\Delta_{ij}^{(l)} = -\eta * g_i^{(l)} * y_j^{(l)}, \quad (2.7)$$

де  $\eta$  – коефіцієнт швидкості навчання ( $0 < \eta < 1$ );

$g$  – градієнт для нейрону  $i$  поточного шару;

$y$  – сигнал нейрону  $j$  з попереднього шару, з яким пов'язаний нейрон  $i$ .

Градієнт залежить від функції помилки, яка, в свою чергу, визначається на основі помилок сигналів, що надходять на виходах нейронів. Помилки на нейронах вихідного шару обчислюються як різниця між фактичними вихідними значеннями та очікуваними сигналами. Таким чином, градієнт для нейрона вихідного шару визначається саме через цю різницю, яка вказує на ступінь відхилення результату мережі від бажаного результату. Ці помилки використовуються для коригування ваг та зміщень в мережі з метою зменшення загальної помилки (формула 2.8).

$$g_j^{(l)} = -\eta * f' (S_j^{(l)}) * \sum_{q=1}^{N^{(l+1)}} w_{jq}^{(l+1)} * q_q^{(l+1)}, \quad (2.8)$$

де  $g_j$  – градієнт нейрону  $j$ ;

$S_j$  – значення суматора цього нейрону;

$f'(x)$  – похідна функції активації;

$y_j$  – отриманий сигнал на нейроні;

$d_j$  – очікуваний сигнал на нейроні.

Для прихованих шарів неможливо безпосередньо визначити критерій якості, тому помилки на нейронах попереднього шару (шару  $L-1$ ) обчислюються як зважена сума градієнтів наступного шару  $L$ . Це дозволяє коригувати ваги нейронів прихованого шару, враховуючи помилки на вихідному рівні та зворотний вплив, що виходить від подальших шарів. Такий підхід забезпечує коректне оновлення параметрів у кожному з шарів і оптимізацію навчання нейронної мережі, а градієнт для цього шару розраховується згідно формули (2.9) за таким принципом:

$$g_j^{(l)} = -\eta * f' (S_j^{(l)}) * \sum_{q=1}^{N^{(l+1)}} w_{jq}^{(l+1)} * q_q^{(l+1)}, \quad (2.9)$$

де  $g_j$  – градієнт нейрону  $j$  прихованого шару  $l$ ;

$\eta$  – коефіцієнт швидкості навчання;

$S_j$  – значення суматора нейрону  $j$ ;  $f'(x)$  – похідна функції активації;

$N$  – кількість нейронів у наступному шарі;

$w_{jq}$  – вага між нейроном  $j$  шару  $l$  та нейроном  $q$  наступного шару;

$g_q$  – градієнт нейрону  $q$  наступного шару.

В залежності від вибраної функції активації, її похідна буде різною. Під час навчання нейронної мережі всі вищезгадані кроки виконуються для всіх ваг і зміщень, по черзі обробляючи всі приклади навчання. Це може включати багаторазове проходження через ті ж самі приклади [39...40]. Як сигнал для зупинки навчання, може виступати досягнення необхідного рівня точності

помилки або встановлений ліміт кількості епох – кількості ітерацій через весь набір навчальних даних.

Функція помилки в нейронній мережі, як і функція активації, вибирається експериментальним шляхом або на основі вже існуючого досвіду в подібних задачах. Для задач класифікації з двома класами зазвичай використовують функції помилки або бінарну крос-ентропію. Важливо зазначити, що бінарна крос-ентропія працює з вихідними значеннями сигмоїдної функції активації, тоді як категоріальна крос-ентропія використовується в поєднанні з функцією softmax, яка зберігає інформацію не лише про максимальні, але й про інші значення вектору виходу (формула 2.10). [41].

$$E = \frac{1}{N} \sum_{i=1}^N (\max(1 - d_i, y_i))^2, y_i \in \{-1; 1\}, \quad (2.10)$$

де  $E$  – помилка нейронної мережі;

$N$  – кількість нейронів у вихідному шарі;

$y_i$  – вихідне значення на  $i$ -му нейроні;

$d_i$  – очікуване значення мережі на  $i$ -му нейроні.

Для класифікації з більш ніж двома класами зазвичай застосовують категоріальну крос-ентропію як функцію помилки нейронної мережі. У задачах регресії використовуються різні функції помилки, серед яких найпоширеніші – середній квадрат помилок, середня сума модулів помилок, середній квадрат логарифмічних помилок та середній абсолютний відсоток помилок. Ці функції помилки дозволяють визначити різницю між передбаченими і реальними значеннями та застосовуються в залежності від типу задачі за допомогою формули (2.11).

$$E = \frac{1}{N} \sum_{i=1}^N (\log(d_i + 1) - \log(y_i + 1))^2, \quad (2.11)$$

де  $d_i$  – фактичне (цільове) значення;

$y_i$  – прогноз моделі;

$\log(d_i+1)$  – стабілізує дисперсію та дозволяє працювати з нульовими числами.

Основною проблемою середнього квадрата помилок є наявність багатьох локальних мінімумів, через які градієнтний спуск може застрягти в одному з них, що значно погіршує якість розпізнавання. Крім того, при використанні цього критерію помилка різко збільшується, якщо є значні, але рідкісні помилки, тому цей критерій варто використовувати лише в тих випадках, коли такі помилки є важливими.

У порівнянні з середнім квадратом помилок, середній модуль помилок зменшує різкість зростання помилки у випадку рідкісних, але значущих помилок. Середній квадрат логарифмічних помилок ще більше знижує вплив таких помилок [42]. В той же час середній абсолютний відсоток помилок дає змогу переходити від абсолютних значень до відсоткових оцінок, що може бути корисно для порівняння ефективності моделі в умовах різних масштабів даних.

Згорткові нейронні мережі складаються із шарів згортки та шарів пулінгу, іноді їх також називають шарами субдискретизації (див. рис. 2.3). Вхідними даними для такої мережі в разі обробки зображень є 2D-тензор, що містить інформацію про яскравість пікселів зображення [43]. Якщо зображення кольорове, воно поділяється на окремі канали RGB, і мережі подаються вже три окремі зображення для кожного з каналів (червоний, зелений, синій) [44].

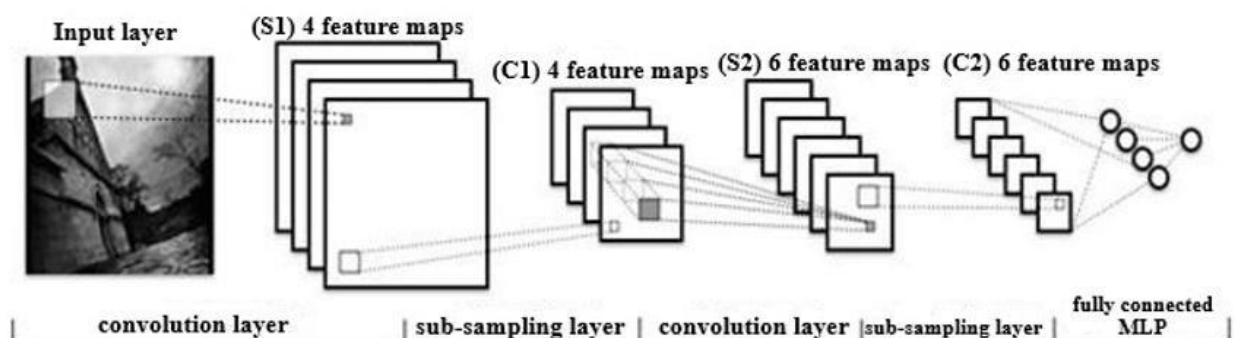


Рисунок 2.3 – Структура згорткової нейронної мережі [44]

Далі для кожного каналу зображення виконується операція згортки з використанням фільтра згортки (див. рис. 2.4). Фільтр має на меті виділити певні ознаки на зображенні, формуючи так звану карту ознак. Наприклад, він може підсилювати горизонтальні чи вертикальні лінії, роблячи їх більш виразними порівняно з іншими елементами зображення [45]. В мережі може бути кілька фільтрів, кожен з яких виділяє різні ознаки на зображенні та створює власну карту ознак.

У випадку двовимірного тензора фільтр являє собою матрицю невеликого розміру, що містить вагові коефіцієнти. Під час виконання операції згортки фільтр рухається по зображенню за допомогою методу ковзаючого вікна. Він множить елементи зображення на відповідні елементи фільтра, потім підсумовує отримані значення та додає ваговий коефіцієнт  $w_0$ , який виконує функцію зсуву. Це значення також проходить через функцію активації і перетворюється на нове значення пікселя на зображенні.

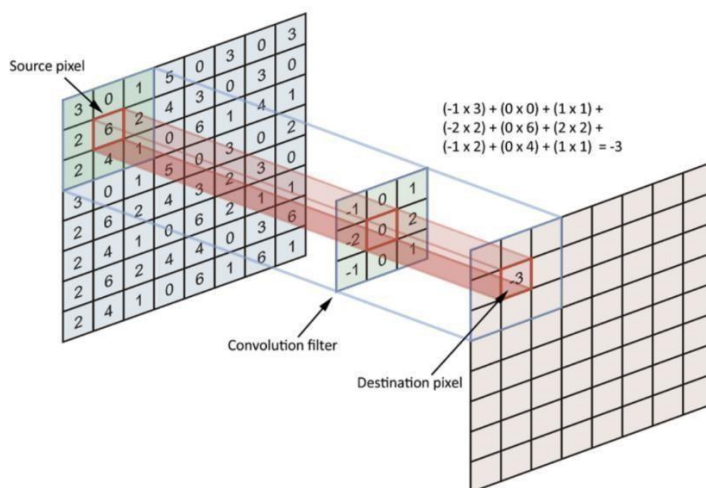


Рисунок 2.4 – Візуалізація процесу згортки [46]

Формальний вираз для розрахунку нового значення пікселя  $v$  за допомогою фільтра розміром  $3 \times 3$  наведено у формулі 2.12.

$$v_{k,m} = f(\sum_{i=1}^3 \sum_{j=1}^3 x_{i+k,j+m} W_{ij} + w_0), \quad (2.12)$$

де  $k, m$  – індекси пікселів на зображенні;

$f$  – функція активації;

$x$  – вихідне значення пікселю;

$w_{ij}$  – відповідна вага у фільтрі згортки;

$w_0$  – зміщення на  $+1$ .

Розмір фільтра та крок, з яким він переміщається по зображенню, є гіперпараметрами мережі, тобто параметрами, які визначаються заздалегідь при налаштуванні мережі і не змінюються під час навчання. Кількість фільтрів також визначається на етапі налаштування і є частиною конфігурації мережі. Кожен фільтр створює свою власну карту ознак (канал), обчислюючи її на основі карт ознак попереднього шару. Якщо на попередньому шарі є кілька каналів, як у випадку з кольоровими зображеннями, фільтр обробляє кожен з цих каналів, і результати потім підсумовуються для створення нової карти ознак.

Зазвичай для фільтру вибирається крок, рівний одиниці. Якщо крок більше одиниці, розмір зображення зменшуватиметься в залежності від значення цього кроку. Після операції згортки зображення стає меншим по краях, оскільки фільтр не може виходити за межі зображення. Щоб уникнути цього ефекту, перед згорткою до зображення додається відступ (*padding*) з нульових значень по краях [46].

Після згортки отримані канали обробляються через шар пулінгу. Шар пулінгу також працює за допомогою ковзного вікна, що проходить по зображенню, і для кожної області вікна обчислюється єдине значення. Існує три типи пулінгу: *max pooling*, *min pooling* і *average pooling*. При *max pooling* вибирається найбільший елемент з вікна, при *min pooling* – найменший, а при *average pooling* – середнє значення. Зазвичай для пулінгових шарів вибирається крок, рівний розміру вікна пулінгу, щоб зображення стискалося і зберігалось тільки найбільш значущі елементи, перед обробкою на наступних шарах мережі. Найпоширенішим є *max pooling*, візуалізацію якого можна побачити на рисунку 2.5, а приклади зміни зображення при його використанні – на рисунку 2.6.

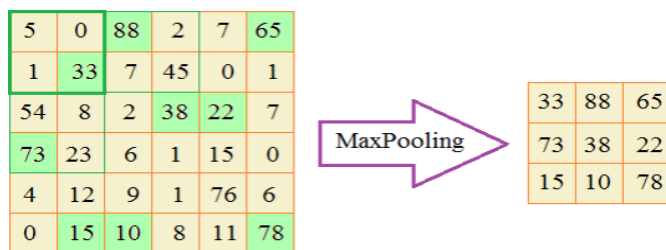


Рисунок 2.5 – Візуалізація процесу max pooling [46]

У згорткових нейронних мережах шари згортки і пулінгу зазвичай чергуються. Кожен новий шар згортки має за мету виділення більш складних ознак зображення. На початкових етапах мережа фокусується на виявленні простих елементів, таких як контури та текстури, а з кожним наступним шаром вона виділяє дедалі більш складні характеристики.



Рисунок 2.6 – Приклади зображення при використанні max pooling [47]



Рисунок 2.7 - Приклади областей, до яких згорткова нейронна мережа більш чутлива в залежності від шару [47]

В дослідженні [47] було вивчено, як нейронні мережі сприймають різні частини зображення, аналізуючи, на які області зображення мережа реагує з найбільшою чутливістю. Приклади таких чутливих областей для різних шарів можна побачити на рисунку 2.7.

Як можна помітити, на початкових шарах згорткової мережі виділяються прості елементи зображення, такі як лінії, градієнти, а також ознаки, що стосуються кольору та відтінку. На другому шарі нейронна мережа починає розпізнавати більш складні форми та текстури. З кожним наступним шаром виділяються все більш складні частини об'єктів, наприклад, колеса машин, а на одному з останніх шарів мережа вже може розпізнавати цілі об'єкти.

Після того як зображення пройшло через шари згортки і пулінгу, утворюються карти ознак, які виділяють найбільш значущі частини зображення у компактному вигляді. Ці карти можуть бути подані на вхід іншим алгоритмам для подальшого вирішення задач, таких як класифікація або регресія. У більшості випадків, після шарів згортки, до нейронної мережі додається повнозв'язаний шар, який відповідає за вирішення конкретної задачі.

Принцип навчання згорткових шарів нейронної мережі подібний до навчання перцептронів. Однак у випадку згорткових мереж оптимізація вагових коефіцієнтів здійснюється за допомогою методу градієнтного спуску, але вже для параметрів, що знаходяться в ядрах згортки.

При розробці глибоких нейронних мереж, що мають велику кількість шарів, виникає проблема загасання і вибуху градієнта. Щоб пояснити, звідки виникають такі ефекти, уявімо, що у нас є мережа з великою кількістю шарів, наприклад, більше ста. Для спрощення припустимо, що функція активації лінійна, а вектор зсувів дорівнює нулю. У такому випадку вихідний вектор мережі буде мати загальний вигляд, наведений у відповідній формулі 2.13.

$$y = w^{[l]}w^{[l-1]}w^{[l-2]} \dots w^{[3]}w^{[2]}w^{[1]}, \quad (2.13)$$

де  $x$  – вхідний вектор мережі;

$w$  – матриця ваг на шарі  $l$ .

Тоді, якщо значення елементів усіх матриць ваг трохи більші, ніж у одиничної матриці, значення будуть рости з кожним наступним шаром, і в результаті значення вихідного вектора  $y$  будуть дуже великими, що призведе до

вибуху градієнта. А якщо значення матриць ваг будуть трохи менші за одиницю, тоді у буде дуже малим, що спричинить загасання градієнта. Оскільки навчання мережі залежить від градієнта, при таких ефектах мережа не зможе ефективно навчатися.

Тому, окрім стандартних повнозв'язних шарів, а також шарів згортки та пулінгу, в глибоких нейронних мережах можуть використовуватися додаткові, комплексні види шарів, що комбінують інші шари. У таких випадках ці шари можуть складати цілі модулі. Одним із таких видів шарів є residual шари (залишкові), які зазвичай використовуються для вирішення проблеми загасання і вибуху градієнта. Значення на нейронах residual шарів формуються так само, як і на звичайному шарі, але з додаванням значень нейронів попереднього шару перед застосуванням функції активації (формула 2.14) [48].

$$a^{[l+2]} = f(w^{[l+2]}a^{[l+1]} + b^{[l+2]} + a^{[l]}), \quad (2.14)$$

де  $a$  – вектор значень на нейронах шару  $l$ ;

$w$  – матриця ваг на шарі  $l$ ;

$b$  – вектор зміщень на шарі  $l$ ;

$f$  – функція активації.

Таким чином, збільшується вплив результату з попередніх шарів на наступні. Було виявлено, що це дозволяє робити нейронні мережі більш глибокими без втрати точності, наприклад, створювати мережі, що складаються з понад сотні або навіть тисячі шарів. Коли в мережі використовуються residual шари, вона отримує назву ResNet, а групи шарів, що реалізують схему, наведену у формулі вище, називаються residual модулями [48].

Також у згорткових мережах іноді виникають труднощі при виборі параметрів фільтра згортки: яку кількість фільтрів вибрати, або, можливо, замість згортки варто застосувати шар пулінгу. Цю проблему вирішує ще один тип комплексних шарів – insertion [49]. Їх ідея полягає в тому, щоб з набору карт ознак, отриманих після попередньої операції, генерувати нові карти ознак за різними

параметрами та об'єднувати їх в одну стопку. Схему цього процесу ілюструє рисунок 2.8.

Таким чином, з попереднього шару можуть виходити кілька відгалужень з власною чередою шарів, кожен з яких має свої параметри, а потім результати цих відгалужень конкатенуються в один шар. У такому випадку мережа під час навчання самостійно вибирає, які параметри та типи шарів є найбільш ефективними. Для непотрібних або менш значущих шарів мережа може обнулити матриці ваг або зробити їх такими малими, що вони матимуть мінімальний вплив на подальші результати. Важливо, щоб розміри результуючих карт ознак збігалися.

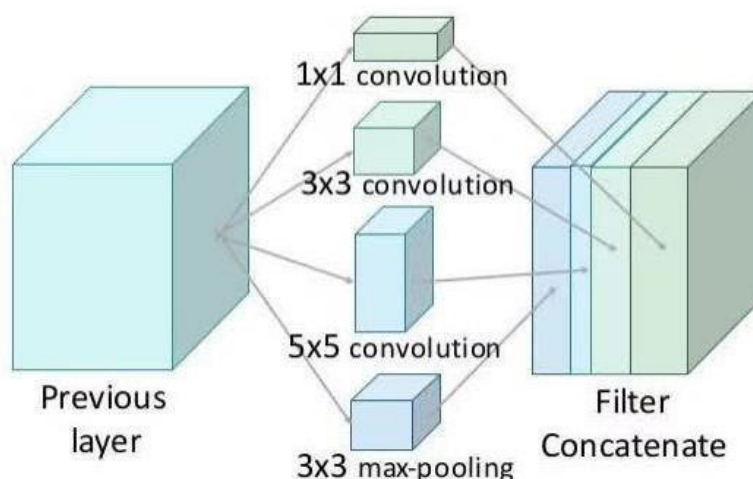


Рисунок 2.8 – Ілюстрація схеми inception модуля [49]

При навчанні методами штучного інтелекту також існує проблема зміщення та дисперсії (bias and variance). У глибокому навчанні вона пов'язана з такими явищами, як недонавчання та перенавчання мережі. Високе зміщення вказує на те, що мережа не може належно підлаштуватися під навчальну вибірку, що призводить до недонавчання, а висока дисперсія показує, що мережа надмірно адаптувалася до навчальної вибірки, що веде до перенавчання. Оптимальним є досягнення балансу між малим зміщенням і дисперсією. Візуалізацію цих ефектів можна побачити на рисунку 2.9.

Точки на кожному графіку представляють навчальні дані, де  $X$  – це вхідні дані для мережі, а  $Y$  – очікувані вихідні значення. При високому зміщенні видно,

що багато значень  $X$  на лінії, яка утворюється прогнозами мережі на тренувальних даних, мають велику відстань до істинно правильних  $Y$ . При високому значенні дисперсії лінія, що утворюється після навчання, проходить через всі навчальні приклади, тому на тестовому датасеті або при подальшій експлуатації мережі, де вхідні приклади мають трохи інші значення, сумарна помилка мережі буде великою [50].



Рисунок 2.9 – Візуалізація проблеми зміщення та дисперсії

На третьому графіку видно, що поєднання низького зміщення та дисперсії дає досить хороший результат – мережа виявляє закономірність між вхідними та вихідними значеннями, не підлаштовуючись під тренувальні дані, тим самим не перенавчаючись.

Зазвичай проблему зміщення та дисперсії вирішують шляхом кращого підбору структури мережі та використання великих датасетів. Однак існують ситуації, коли неможливо покращити структуру мережі або використовувати більший датасет. У таких випадках застосовують метод регуляризації. Існує два основні види регуляризації: L2 та L1.

L2-регуляризація реалізується шляхом додавання до функції помилки додаткового члена, що веде до утворення нової функції помилки, яку можна подати в формулі 2.15.

$$E = \frac{1}{m} \sum_{i=1}^m L(d_i, y_i) + \frac{\lambda}{2m} \sum_{i=1}^n \|w^{[l]}\|_2^2, \quad (2.15)$$

де  $m$  – кількість навчальних прикладів;

$L$  – використовувана функція помилки;

$\lambda$  – гіперпараметр L2-регуляризації;

$n$  – кількість шарів мережі, виключаючи вхідний шар;

$w$  – матриця ваг шару  $l$ .

L1-регуляризація дозволяє «стискати» матрицю ваг, додаючи до неї більше нулів. Оскільки нулі займають менше пам'яті, ваги також займають менше місця. Однак практика показує, що це дозволяє зменшити зайняте місце лише незначно, тому значно частіше використовується L2-регуляризація.

Параметр  $\lambda$  називається параметром регуляризації і підбирається експериментальним чином в межах від 0 до нескінченності. Цей параметр налаштовує чутливість до вхідних даних. Чим більше значення  $\lambda$ , тим менш чутливими до вхідних значень стають прогнози мережі. Зазвичай оптимальним значенням цього параметра є одиниця.

Комбінуючи вищеописані методи, можна робити нейронні мережі дуже глибокими та значно ефективнішими за якістю розпізнавання, ніж звичайні мережі, що не використовують ці методи.

## 2.2 Архітектура системи автоматичного розпізнавання обличчя

Система розпізнавання обличчя складається з кількох етапів: детекція людських образів на зображенні, виділення ознак кожного обличчя та їх безпосереднє розпізнавання [51].

Детекція обличчя, як і детекція будь-якого іншого об'єкта на зображенні, полягає в знаходженні координат його області, що його обмежує (bounding box) в пікселях. За допомогою нейронної мережі, ця задача може вирішуватися як завдання регресії, в якому потрібно знайти ці координати [52].

На перший погляд, можна навчити мережу, яка на вхід приймає зображення, а на виході видає чотири числа – координати центру або одного з кутів знайденої області, а також її висоту і ширину. Це найбільш очевидний спосіб, однак його недоліком є те, що він в кращому випадку зможе знаходити лише один об'єкт. На

різних зображеннях може бути різна кількість осіб, а архітектура мережі задається спочатку і не змінюється, тому потрібен інший підхід.

Таким підходом, що в тому чи іншому вигляді використовується у всіх мережах детекції, є реалізація детекції через класифікацію з використанням ковзаючого вікна. Воно проходить по зображенню і класифікує те, що знаходиться в області вікна. Крок і розмір ковзаючого вікна можуть вибиратися будь-яким, часто прохід по зображенню виконується кілька разів з різним розміром вікна або зображення.

Ця ідея працює добре, однак вона може містити величезну кількість областей, кожен з яких необхідно класифікувати, а тому процес може бути дуже повільним. Для вирішення такої проблеми виділяють два основних типи методів: двоетапні і одноетапні.

Двоетапні методи на першому етапі за допомогою будь-якого допоміжного алгоритму знаходять регіони інтересу, де можуть знаходитись об'єкти, а потім на другому етапі ці регіони перевіряються класифікатором з локалізацією. Локалізація уточнює координати об'єкта в області. Існує безліч алгоритмів знаходження регіонів інтересу, порівняння деяких з яких наведено в таблиці 2.1.

Таблиця 2.1 – Методи пропозиції регіонів інтересу

Метод	Підхід	Сегменти виходів	Оцінка результатів	Контроль пропозицій	Час (с.)	Повторюваність	Результати відклику	Результати виявлення
Bing	Windows scoring		✓	✓	0.2	★★★	★	
CPMC	Grouping	✓	✓	✓	250		★★	★
EdgeBoxes	Windows scoring		✓	✓	0.3	★★	★★★	★★★
Endres	Grouping	✓	✓	✓	100		★★★	★★
Geodesic	Grouping	✓		✓	1	★	★★★	★★
MCG	Grouping	✓	✓	✓	30	★	★★★	★★★
Objectness	Windows scoring		✓	✓	3		★	
Rahtu	Windows scoring		✓	✓	3			★
RandomizedPrim's	Grouping	✓		✓	1	★	★	★★
Rantalankila	Grouping	✓		✓	10	★★		★★
Rigor	Grouping	✓		✓	10	★	★★	★★
SelectiveSearch	Grouping	✓	✓	✓	10	★★	★★★	★★★

Зазвичай використовується метод селективного пошуку (Selective Search) або метод Edge Boxes, оскільки вони працюють досить швидко і дають добрі

результати детекції. Таким чином, зменшується кількість областей для перевірки і прискорюється загальна швидкість роботи системи.

Одноетапні методи ґрунтуються на використанні згорткової реалізації ковзаючих вікон за допомогою згорткових нейронних мереж [53]. Її суть в тому, що після проходу зображення по шарах згортки, кожен піксель на результуючій карті ознак відповідає області пікселів на оригінальному зображенні (рис. 2.11).

Згорткова нейронна мережа виконує всі ті ж лінійні перетворення, а значить, можна навчити її виконувати завдання регресії таким чином, щоб кожна з останніх карт ознак відповідала певному значенню, як вихідні нейрони в повнозв'язній мережі. Так, перша і друга карти ознак можуть визначати координату центру області, а третя і четверта – її ширину і висоту. Додатково до координат області в такому методі додається ще значення, що визначає ймовірність або впевненість мережі в тому, що в даній області є шуканий об'єкт.

Оскільки згортка сама по собі реалізується як ковзаючі вікна, то такий механізм вже закладений у згорткові мережі, що дає значний приріст у швидкості.

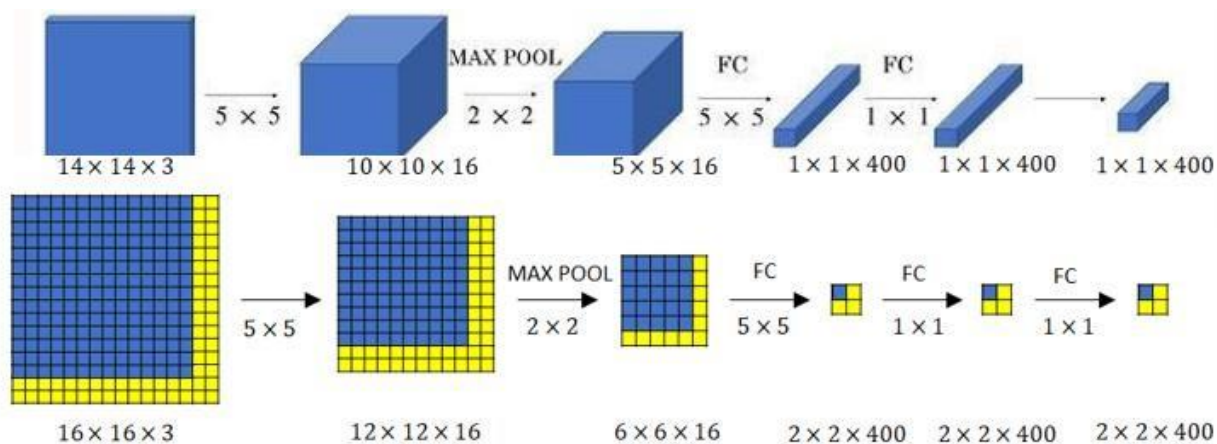


Рисунок 2.10 – Приклад та візуалізація конволюційного підходу детекції об'єктів за допомогою згорткових нейронних мереж [50]

На практиці, після виконання класифікації всіх необхідних областей на зображенні, нейронна мережа може виділити велику кількість таких прямокутних областей, які можуть перетинатися одна з одною та вказувати на однакові об'єкти. Однак, зрештою, потрібно визначити лише одну область на одне обличчя. Тому

після визначення областей, які містять обличчя людей, застосовується метод придушення областей, у яких упевненість присутності обличчя не є максимальною серед усіх знайдених областей. Цей метод називається Non Maximum Suppression (NMS) [54].

Алгоритм цього методу полягає в тому, що з усіх детектованих областей вибирається лише та, що має максимальний коефіцієнт впевненості, виданий мережею. Однак потрібно також враховувати, що на зображенні можуть бути розташовані кілька об'єктів, які детектуються, при цьому відстань між ними буде невеликою, тому потрібно розрізнити знайдені області за тим, до яких об'єктів вони відносяться. Для цього перед вибором області з максимальною впевненістю кожна область порівнюється з іншими, і для кожних двох областей обчислюється значення метрики Intersection Over Union (IOU). Ця метрика використовується для порівняння, наскільки області перетинаються одна з одною. Вона розраховується як площа перетину, поділена на площу об'єднаної області. Більше значення показує більшу міру перетину, причому значення, рівне одиниці, означає, що області повністю збігаються, а значення, рівне нулю, – що вони не перетинаються.

Після порівняння ступеня перетину всіх областей, вони кластеризуються за допомогою введення порогового значення. Це значення є гіперпараметром, який підбирається вручну для досягнення найкращих показників точності детекції. Якщо ступінь перетину між двома областями перевищує поріг, вважається, що вони вказують на той самий об'єкт, і тому вони групуються разом у кластер, з якого потім вибирається область з найбільшою впевненістю.

Крім цього методу, існують й інші, які є його продовженням з певними модифікаціями для досягнення кращих результатів точності детекції, такі як Soft NMS, Softer NMS, IOU-Guided NMS, Adaptive NMS, DIOU-NMS, кожен з яких має свої переваги та недоліки.

Після детекції обличчя на зображенні для кожного обличчя виділяються ознаки, які унікально його характеризують у стислому вигляді. Такі ознаки можуть бути представлені у вигляді вектора чисел. При використанні глибокого навчання, ознаки виділяються самими нейронними мережами. ЗНМ виділяють ознаки у

вигляді результуючих карт ознак зображення, а якщо в якості другої частини згорткової мережі додається перцептрон, то вектор ознак – вихідний вектор мережі.

Безпосереднє розпізнавання людських образів, у свою чергу, ділиться на дві категорії: ідентифікація та верифікація. Ідентифікація обличчя – це процес порівняння одного обличчя з усіма розміченими зображеннями людських образів, які містяться в базі даних (БД), з метою знайти найбільш схоже або прийняти рішення про те, що такого обличчя в базі даних немає [55]. Під розміченими зображеннями мається на увазі, що разом із зображенням обличчя в БД зберігається мітка, яка ідентифікує людину, наприклад, його ім'я. Верифікація ж виконується тільки між двома зображеннями людських образів, щоб прийняти рішення – чи це одна й та сама людина, чи ні.

Для вирішення завдання ідентифікації та верифікації вектори ознак, отримані з двох зображень, порівнюються між собою. Це часто здійснюється за допомогою сіамських нейронних мереж. Сіамські нейронні мережі складаються з двох однакових нейронних мереж, як за структурою, так і за навченими вагами, тобто по суті це два екземпляри однієї й тієї ж мережі. В контексті задачі розпізнавання зображень, на вхід кожної з мереж подається одне зображення, а на виході кожна мережа видає вектор чисел, який стисло представляє це зображення. Кожній мережі подається своє зображення, а завдання всієї структури сіамської нейронної мережі – порівняти вектори на виходах двох її мереж і зробити висновок про їх схожість. Для прийняття безпосереднього рішення про схожість використовується додатковий пороговий гіперпараметр.

Якщо різниця між векторами менша за заданий поріг, можна зробити висновок, що на зображеннях представлений один і той самий об'єкт або об'єкт одного й того ж класу. Для ідентифікації та верифікації обличчя в якості гіперпараметра встановлюється граничне значення схожості, при якому вважається, що на зображеннях знаходиться одна й та сама людина.

Саму мережу, що кодує зображення у вектори ознак, можна навчати двома способами: навчати безпосередньо метрику схожості векторів або виділяти ознаки через навчання класифікатора [57].

Навчання метрики схожості полягає в тому, щоб мережу навчити таким чином, щоб вектори ознак для зображень людських образів однієї й тієї ж людини були близькі один до одного по відстані або куту, а для різних людей – суттєво різнилися, щоб можна було з упевненістю прийняти рішення про те, що ці вектори не схожі.

Метод виділення векторів ознак із зображень людських образів через навчання класифікатора полягає в тому, що навчається нейронна мережа, яка класифікує зображення обличчя між фіксованим набором людей, після чого просто видаляється останній шар, що класифікує. Вся мережа, що залишилась, вже вміє виділяти ознаки.

У такому підході головним завданням стає правильний вибір функції помилки. Зазвичай для класифікації використовується функція помилки Softmax, але в контексті розпізнавання осіб вона має великий недолік – дана функція помилки не дає чіткого поділу об'єктів, що класифікуються, між їх класами. Тому, якщо об'єкт одного класу має велику схожість з об'єктами іншого класу, то при візуалізації цієї задачі як кластеризації, він буде знаходитися на краю кластера, в результаті чого мережа може легко віднести об'єкт до невірному класу.

### 2.3 Висновки

У даному розділі було обґрунтовано вибір нейронних мереж як основного інструмента для реалізації системи ідентифікації. Надано теоретичну базу щодо структури, роботи та навчання штучних нейронних мереж, зокрема персептронів, функцій активації та принципів оптимізації моделей. Особлива увага приділена згортковим нейронним мережам, які найкраще підходять для обробки зображень. Також розглянуто архітектуру майбутньої системи, що дозволяє обробляти дані в реальному часі та забезпечує масштабованість і точність розпізнавання.

## 3 РОЗРОБКА МОДЕЛІ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ

### 3.1 Огляд сучасних згорткових мереж для розпізнавання

Моделі сучасних нейронних мереж за своїм практичним призначенням умовно поділяють на два великі класи: мережі для детекції та мережі для ідентифікації або верифікації об'єктів. Кожен із цих напрямів має свою особливу історію розвитку, внутрішню логіку побудови та відмінності у застосуванні [58]. І першим нашим кроком на шляху розуміння стане знайомство з мережами для детекції об'єктів, які відіграють фундаментальну роль у багатьох сучасних технологічних рішеннях – від автономних автомобілів до систем відеоспостереження і безпеки.

Серед яскравих представників цієї групи можна виділити три основні родини моделей: мережі сімейства R-CNN (R-CNN, Fast R-CNN, Faster R-CNN), надзвичайно популярні YOLO (You Only Look Once) та універсальну і водночас витончену архітектуру MTCNN (Multi-task Cascaded Convolutional Networks).

R-CNN – справжня класика жанру, яка започаткувала нову еру в підходах до задач детекції. Перший варіант цієї мережі – R-CNN (Regions with Convolutional Neural Networks) – належить до так званих двохетапних методів детекції. Це означає, що процес обробки зображення поділений на два чіткі етапи: пошук потенційних регіонів, де ймовірно знаходяться об'єкти, та подальша їхня класифікація.

Алгоритм роботи R-CNN можна описати наступним чином: спочатку модель генерує велику кількість потенційних областей (понад дві тисячі регіонів для одного зображення) за допомогою алгоритму Selective Search. Далі кожен із цих регіонів масштабується до однакового розміру і передається через згорткову нейронну мережу для витягання ознак (features). Нарешті, класифікатор SVM (Support Vector Machine) приймає рішення про те, чи містить дана область певний об'єкт, і якщо так – який саме.

Однак незважаючи на високу точність, така архітектура мала один суттєвий недолік: час обробки. Навіть із сучасними обчислювальними потужностями повна

обробка одного зображення займала кілька секунд – надто повільно для застосувань у реальному часі.

Щоб вирішити цю проблему, з'явилася Fast R-CNN. Головною інновацією стала ідея: замість проганяти кожен регіон через окрему згорткову мережу, зображення повністю проходить через нейронну мережу один раз, і лише після цього, на рівні ознак, відбувається визначення об'єктів. Це дозволило суттєво пришвидшити обробку без втрати точності. Більш того, Fast R-CNN ввела концепцію ROI Pooling (Region of Interest Pooling), яка ще більше оптимізувала процес обробки регіонів.

Ще більший стрибок уперед забезпечила Faster R-CNN – модель, що ввела RPN (Region Proposal Network), окрему нейронну мережу для швидкої генерації регіонів інтересу без необхідності застосовувати Selective Search. Це дозволило системі працювати в реальному часі на потужних GPU, прокладаючи шлях до використання нейронних мереж у мобільних додатках, відеоаналітиці та навіть автономному транспорту.

На рисунку 3.1 наведено спрощену схему роботи алгоритму класичної моделі R-CNN, де чітко видно поділ процесу на генерацію регіонів, екстракцію ознак та класифікацію.

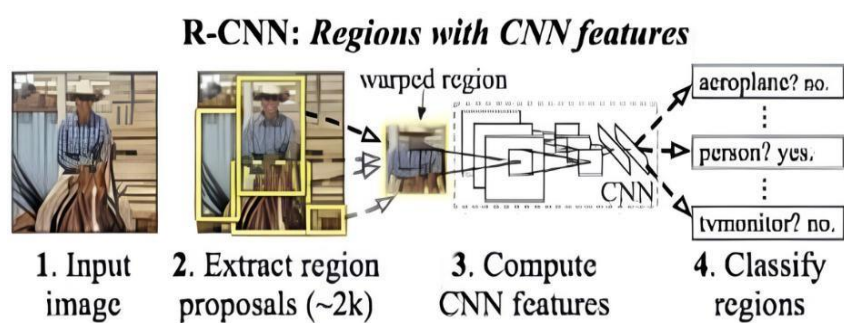


Рисунок 3.1 – Алгоритм роботи R-CNN [60]

У той час як родина R-CNN послідовно вдосконалювала точність і методологію двохетапної детекції, у світі штучного інтелекту виникла нова парадигма: "читай усе зображення одним поглядом". Так народилася модель YOLO (You Only Look Once).

На відміну від своїх попередників, YOLO обробляє зображення цілісно, розділяючи його на сітку і прогнозуючи для кожної комірки імовірність наявності об'єкта та його координати. Цей підхід забезпечує фантастичну швидкість – десятки кадрів за секунду – і робить YOLO незамінною для задач, де критично важлива обробка в реальному часі: в автоматизованих виробничих лініях, у дронах, у відеоіграх та кібербезпеці.

Щоправда, варто відзначити: класична версія YOLO дещо поступалася моделям на базі R-CNN у точності локалізації дрібних об'єктів. Однак завдяки подальшим ітераціям (YOLOv2, YOLOv3, YOLOv4 і, зрештою, сучасні версії YOLOv8) ці недоліки були поступово подолані.

Окремої уваги заслуговує модель MTCNN – елегантне рішення для виявлення обличчя у зображеннях та відео. Вона працює за принципом каскадного розпізнавання: кілька послідовних нейронних мереж уточнюють положення обличчя на зображенні, одночасно визначаючи ключові точки – очі, ніс, кути губ. Саме завдяки цій моделі мобільні телефони навчилися розблоковуватися за допомогою розпізнавання облич у долі секунди.

Унікальність MTCNN полягає у її здатності обробляти зображення на різних масштабах за допомогою трьох послідовних етапів (P-Net, R-Net і O-Net), що забезпечує виняткову точність навіть на складних або зашумлених зображеннях.

Регіони інтересу в класичній мережі R-CNN визначаються за допомогою методу селективного пошуку (Selective Search) – на той час справжнього прориву у завданні генерації кандидатних областей. Принцип його роботи базується на аналізі базових властивостей зображення: кольору, текстури, форми та відтінків. По суті, алгоритм шукає ділянки, в межах яких характеристики пікселів є подібними, а отже, там із високою ймовірністю може знаходитись певний об'єкт.

Процес виглядає так: спочатку зображення розбивається на дрібні сегменти, які потім поступово об'єднуються за певними критеріями подібності. На кожному кроці формується набір потенційних регіонів інтересу (Region Proposals) – зазвичай їх виходить кілька тисяч для одного зображення. І хоча така техніка дозволяла уникати повного перебору всіх можливих прямокутників (що математично

неможливо для великих зображень), її продуктивність усе ще залишала бажати кращого.

Після генерування регіонів, кожен із них окремо подавався через згорткову нейронну мережу для витягання ознак (features) – векторів, що описують вміст регіону у вигляді компактного числового представлення. Ці ознаки передавалися далі до методу опорних векторів (SVM) для класифікації: чи є в регіоні об'єкт певного класу чи ні [61]. Паралельно застосовувалась окрема регресійна модель, яка уточнювала координати об'єкта в межах виявленого регіону.

Ця багатоступенева схема мала свої очевидні переваги: висока точність детекції для об'єктів різного розміру та форми, адаптивність до складних сцен. Але, як це часто буває у світі інновацій, з перевагами прийшли й недоліки.

Головна проблема полягала в тому, що кожен регіон оброблявся окремо – і це створювало величезне навантаження на обчислювальні ресурси. Виділення ознак, виконання класифікації та регресії для кожної області займали багато часу навіть на найсучасніших тоді графічних процесорах. В умовах зображення з тисячами регіонів процес обробки розтягувався на хвилини, що робило таку систему малоприсадною для реального часу.

Більше того, класифікаційна та регресійна моделі у R-CNN були розділеними. Їх потрібно було тренувати окремо, що призводило до значного збільшення часу навчання і вимог до обсягу пам'яті для зберігання параметрів усіх цих моделей. Словом, перша версія R-CNN була чудовою концепцією, але її практичне застосування потребувало радикальної оптимізації.

Щоб подолати ці фундаментальні обмеження, дослідники запропонували нову архітектуру – Fast R-CNN. Ця модель стала символом наступного етапу розвитку в детекції об'єктів [62].

Основна інновація Fast R-CNN полягала у принциповій зміні порядку обробки: замість того, щоб подавати кожен регіон окремо через згорткову мережу, спочатку обробляється усе зображення цілком. На виході формується карта ознак (feature map) – багатовимірне представлення просторових характеристик усього зображення.

Далі застосовується особливий механізм, відомий як ROI Pooling (Region of Interest Pooling). Кожен запропонований регіон інтересу "накладається" на карту ознак, і з відповідної області витягується фіксованого розміру вектор ознак. Цей процес суттєво зменшує обчислювальні витрати, оскільки згорткові операції виконуються лише один раз на всю картинку, а не для кожного регіону окремо.

Ще одна революційна ідея Fast R-CNN – об'єднання класифікації та регресії в єдину повнозв'язну нейронну мережу із двома вихідними шарами. Один вихід передбачає клас об'єкта (або фон), а інший вихід уточнює координати об'єкта у просторі зображення.

Результати були вражаючими: Fast R-CNN була у декілька разів швидшою за свого попередника, зберігаючи або навіть покращуючи точність детекції. Вона змогла обробляти великі об'єми даних практично в реальному часі, відкривши шлях для використання нейронних мереж у таких сферах, як онлайн-відстеження об'єктів, безпілотне водіння, системи контролю доступу та розумне відеоспостереження.

Архітектуру Fast R-CNN, з її ключовими етапами – обробкою всього зображення, витяганням регіонів через ROI Pooling, спільною нейронною мережею для класифікації та регресії – можна побачити на рисунку 3.2.

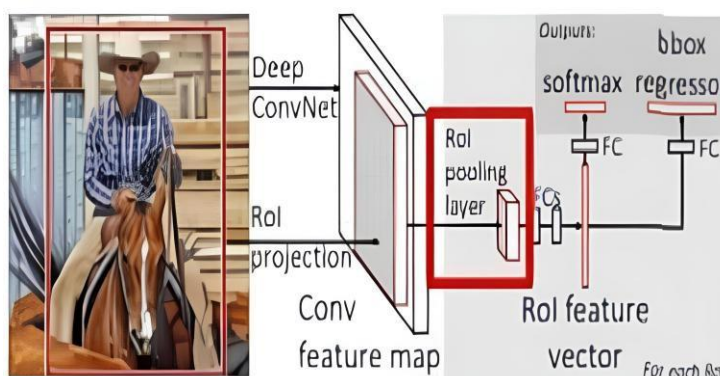


Рисунок 3.2 – Архітектура мережі Fast R-CNN [63]

Подальший розвиток мережі Fast R-CNN ознаменував нову віху у галузі комп'ютерного бачення – на сцену вийшла Faster R-CNN. Ця модель стала

справжнім проривом, оскільки вирішила одне з найболючіших місць своїх попередників: тривалий та ресурсоємний етап визначення регіонів інтересу [63].

Якщо раніше для цієї задачі застосовувався метод селективного пошуку, який хоч і забезпечував пристойну точність, але значно уповільнював обробку, то в Faster R-CNN все було змінено докорінно. Від селективного пошуку відмовилися на користь спеціально розробленої мережі під назвою Region Proposal Network (RPN).

RPN – це окрема згорткова нейронна мережа, яка навчається генерувати пропозиції регіонів безпосередньо на основі карти ознак зображення. Таким чином, зображення проходить через згорткову частину один раз, і вже на основі отриманої карти ознак генерується безліч регіонів потенційного розташування об'єктів. Це рішення не лише радикально прискорило процес, а й забезпечило вищу узгодженість між пошуком регіонів і подальшою їх обробкою, оскільки обидва етапи почали використовувати спільні ознаки.

Інновація полягала й у тому, що концепція "якорів" (anchor boxes) дозволила значно підвищити гнучкість системи: тепер мережа могла одночасно виявляти об'єкти різних розмірів і пропорцій у межах одного регіону. Кожен якор відповідав за певний масштаб і співвідношення сторін, що дозволяло покривати більшу різноманітність можливих об'єктів на сцені.

Паралельно зі вдосконаленнями Faster R-CNN розвивалися й інші підходи до детекції об'єктів, серед яких мережа YOLO (You Only Look Once) зайняла особливе місце. Філософія YOLO кардинально відрізняється: замість того, щоб спочатку знаходити регіони, а потім їх класифікувати, ця мережа намагається вирішити обидві задачі одночасно за один єдиний прохід через нейронну мережу.

У YOLO зображення ділиться на сітку рівномірних осередків. Кожен осередок відповідає за виявлення об'єктів, центр яких знаходиться в його межах. Модель одночасно передбачає координати обмежуючих рамок, клас об'єкта та рівень впевненості для кожного передбачення. Завдяки такому підходу YOLO досягає неймовірної швидкості – саме тому вона стала стандартом для реального часу в задачах відеоспостереження, автопілотування та мобільних додатків [64].

Однак, щоб досягти високої точності при одночасній швидкості, YOLO також впроваджує концепцію якірних рамок, аналогічно до Faster R-CNN. При цьому мережа навчається одночасно розпізнавати кілька об'єктів у кожному осередку сітки. Різні варіанти моделей YOLO – такі як YOLOv2, YOLOv3 та новітніші версії, як YOLOv5 і YOLOv8 – продовжили удосконалювати цю архітектуру, додаючи нові механізми для поліпшення як точності, так і продуктивності.

Крім одноетапних та двоетапних підходів до детекції об'єктів, паралельно розвивалися й каскадні методи, що використовують послідовність мереж для досягнення найкращих результатів. Одним із найяскравіших представників такого підходу стала мережа MTCNN (Multi-task Cascaded Convolutional Networks).

MTCNN спеціалізується на детекції людських образів на зображеннях та відео і демонструє надзвичайно високу ефективність завдяки багаторівневій структурі (рисунок 3.3).

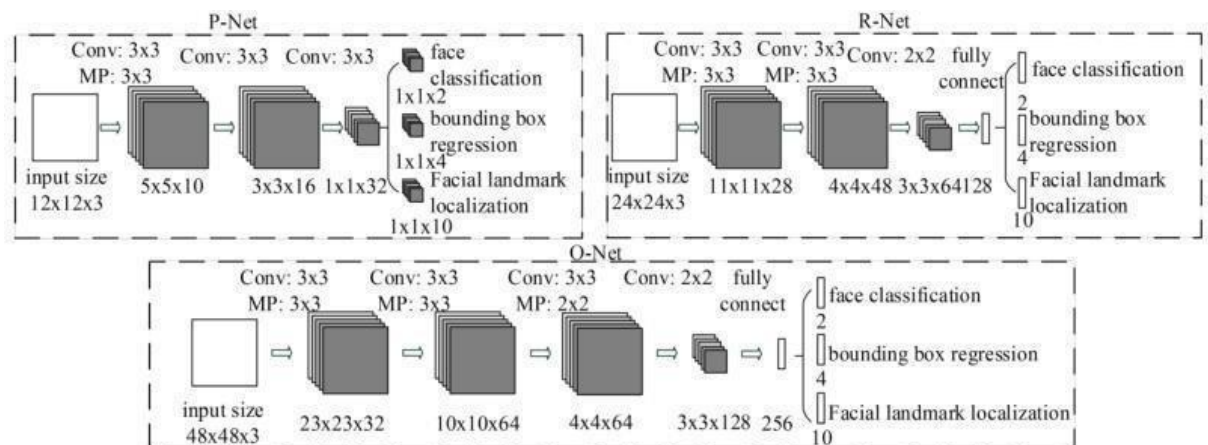


Рисунок 3.3 – Архітектура мережі MTCNN [64]

MTCNN складається з трьох згорткових мереж, що працюють послідовно:

P-Net (Proposal Network) відповідає за грубе виявлення кандидатів на обличчя та первинну регресію обмежуючих рамок.

R-Net (Refine Network) відфільтровує помилкові позитивні результати, залишаючи лише найбільш перспективні варіанти.

O-Net (Output Network) уточнює місцезположення облич, визначає п'ять основних ключових точок обличчя (очі, кінчик носа та кути рота) і вирівнює обличчя для подальшої обробки.

Особливу увагу в MTCNN приділено саме точності вирівнювання обличчя, що критично важливо для таких завдань, як розпізнавання осіб або емоційний аналіз.

Процес роботи мережі починається з побудови піраміди зображень, що дозволяє ефективно знаходити обличчя різних розмірів. Кожне масштабоване зображення проходить через каскадну систему, де на кожному етапі підвищується точність детекції і зменшується кількість кандидатів, що значно оптимізує як швидкість обробки, так і споживання ресурсів.

Загалом еволюція моделей від R-CNN до Faster R-CNN, YOLO та MTCNN віддзеркалює глибинну зміну в підходах до задач комп'ютерного бачення. Від повільних, але точних систем ми перейшли до рішень, здатних працювати в режимі реального часу без суттєвих компромісів у якості. Кожен новий етап цієї еволюції відкриває нові можливості для застосування нейронних мереж у повсякденному житті: від безпілотних автомобілів до систем відеоспостереження, медичної діагностики та доповненої реальності.

Згідно з викладеним вище, варто наголосити: між швидкістю роботи нейронної мережі та її точністю існує тонкий, делікатний баланс, від якого залежить успіх застосування тієї чи іншої моделі в реальних завданнях розпізнавання. Це співвідношення виявляється особливо важливим у контексті сучасних досліджень у галузі комп'ютерного зору, де кожна мілісекунда має значення, а кожна похибка може стати фатальною в задачах безпеки чи автоматизації [65].

В одному з авторитетних досліджень, присвячених цій проблематиці, було здійснено ґрунтовне порівняння одноетапних і двоетапних методів детекції об'єктів. Для цього взяли за основу популярний набір даних COCO (Common Objects in Context) – справжню перлину у світі штучного інтелекту. Цей датасет охоплює близько 330 тисяч зображень, на яких розміщено понад півтора мільйона

об'єктів, що належать до 80 різних категорій – від людей і тварин до меблів і транспортних засобів. Завдяки такій різноманітності COCO став золотим стандартом для тренування і тестування алгоритмів розпізнавання та сегментації об'єктів.

Щоб проілюструвати результати, у дослідженні було побудовано графік (рисунок 3.4), який став справжньою картою навігації в океані нейронних мереж. По осі X відкладено час, необхідний для обробки одного зображення, виміряний у мілісекундах, а по осі Y – середня точність (mAP, mean Average Precision) в процентах. З першого погляду на цей графік стає очевидним: моделі, що працюють за одноетапною схемою, наприклад YOLO чи SSD, демонструють неймовірну швидкість, проте поступаються у точності своїм двоетапним колегам, серед яких особливе місце займає Faster R-CNN. Двоетапні методи, у свою чергу, забезпечують вищу якість розпізнавання, але вимагають більше часу на обробку кожного кадру, що може бути критичним у реальних часозалежних застосуваннях.

Ця дилема "швидкість проти точності" довгий час залишалася одним із головних викликів для дослідників. Проте з'явилася інноваційна робота, яка запропонувала блискучий компроміс – це розробка нейронної мережі RetinaNet. Її творці підійшли до проблеми фундаментально: вони зрозуміли, що основною причиною нижчої точності одноетапних моделей є сильна розбалансованість прикладів – коли на зображеннях переважають фонові області без об'єктів, і лише мала частка прикладів містить цільові об'єкти.

Щоб усунути цю проблему, RetinaNet впровадила нову функцію втрат – Focal Loss. Вона розумно зменшує вагу легких прикладів (тобто правильних прогнозів), фокусуючи увагу моделі на складних випадках, де об'єкт важко відрізнити від фону. Завдяки такій інновації RetinaNet зуміла поєднати найкраще з обох світів: зберегти шалену швидкість, характерну для одноетапних підходів, і досягти рівня точності, який раніше був притаманний лише важким і повільним двоетапним системам.

Особливо вражає те, що RetinaNet стала майже універсальним рішенням для завдань реального часу від відеоспостереження до автономного водіння. Її

здатність працювати на різних роздільних здатностях і легко адаптуватися до нових датасетів зробила її справжнім проривом у галузі.

Важливо також зазначити, що розвиток функцій втрат, таких як Focal Loss, не лише покращив якість одноетапних моделей, а й сприяв переосмисленню традиційних уявлень про те, як нейронні мережі повинні навчатися (формула 3.1).

Підводячи підсумок, можна сказати, що сьогодні ми стоїмо на межі нової епохи у світі комп'ютерного зору, де завдяки таким рішенням, як RetinaNet, питання вибору між швидкістю і точністю перестає бути дилемою. І майбутнє обіцяє ще більше захопливих відкриттів, у яких баланс, витонченість і ефективність стануть головними рушійними силами прогресу.

$$FL(p, \gamma) = -(1 - p_t)^\gamma \log(p_t), p_t = \begin{cases} p, & \gamma = 1 \\ 1 - p, & \text{інакше} \end{cases}, \quad (3.1)$$

де  $p$  – передбачений мережею результат;

$\gamma$  – очікуваний результат;

$\gamma$  – гіперпараметр згладжування, чим більше значення, тим більш чутливою буде помилка до прикладів, складних для розпізнавання.

На рисунку 3.4 показано графік залежності детекції від швидкості.

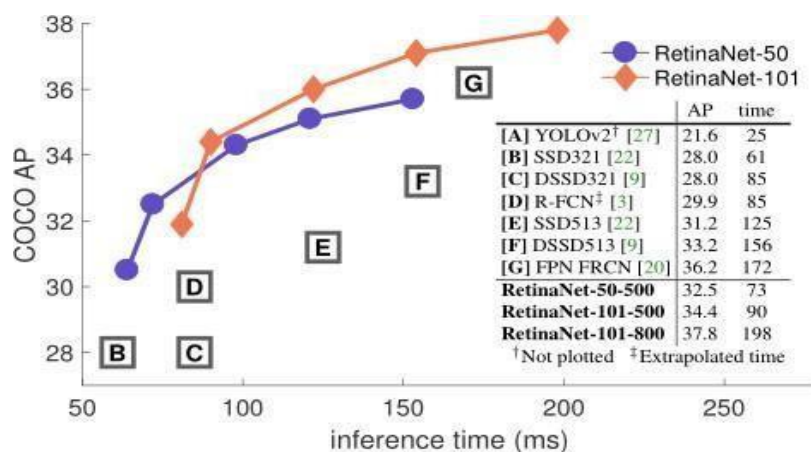


Рисунок 3.4 – Залежність точності детекції від швидкості

Серед найяскравіших здобутків у сфері розпізнавання облич особливе місце займає модель FaceNet. Вона стала справжньою революцією, відкривши нові

горизонти для ідентифікації та верифікації осіб у різноманітних додатках – від систем безпеки до соціальних мереж.

FaceNet побудована на основі так званої сіамської нейронної мережі. Її робота нагадує витончену взаємодію двох глибоких мереж-близнюків: кожне з двох вхідних зображень проходить крізь власну копію нейронної мережі, яка перетворює обличчя на компактний числовий вектор ознак. Цей вектор – своєрідний "відбиток" обличчя у багатовимірному просторі. Далі система порівнює отримані вектори: якщо вони розташовані близько один до одного, то з великою ймовірністю це одна й та сама людина; якщо ж між ними велика відстань, то це означення, що перед нами різні особи.

Однак не менш важливим є не лише сам механізм перетворення обличчя у вектори, а й спосіб навчання мережі. FaceNet поставила собі амбітну мету: створити такі векторні представлення, які були б максимально однаковими для одного обличчя за будь-яких змін, чи то інший кут зйомки, чи інше освітлення, чи навіть змінена міміка. І водночас вектори різних людей повинні суттєво відрізнятися, щоб уникнути помилкових збігів.

Для досягнення цієї цілі була розроблена спеціальна функція втрат під назвою Triplet Loss. Її принцип можна описати доволі простою і водночас дуже ефективною ідеєю: навчання здійснюється не на основі окремих зображень чи пар, а на основі трійок прикладів. Для кожного кроку навчання обирається три зображення: "якір" – базове фото певної особи; "позитивний приклад" – інше фото тієї ж людини; і "негативний приклад" – фото зовсім іншої людини.

Мережа навчається таким чином, щоб вектор ознак якоря був як найближчим до вектора позитивного прикладу та як найдалі від вектора негативного прикладу. Іншими словами, система вчиться інтуїтивно розрізняти, що "свій" має бути ближчим, а "чужий" – віддаленим. Причому ця відстань має бути достатньо великою, щоб зберігати чітке розмежування між різними особами навіть у складних випадках.

Особливу роль у цьому процесі відіграє правильний підбір трійок для навчання. Якщо обирати надто прості приклади, де обличчя різних людей сильно

відрізняються, то мережа не навчиться тонко розпізнавати схожі обличчя. Якщо ж одразу подавати надто складні трійки, де різні люди виглядають майже однаково, навчання може стати нестабільним. Тому у FaceNet використовується спеціальна стратегія вибору так званих "складних негативних прикладів" таких образів, які зовні нагадують якоря, але належать іншим особам.

Завдяки такому підходу FaceNet досягає надзвичайної точності. Ще одним важливим аспектом є те, що результат роботи мережі – це не фіксоване рішення "так/ні", а вектор ознак, який можна зберігати, порівнювати, об'єднувати. Це робить систему неймовірно гнучкою: для додавання нової особи до бази не потрібно перенавчати мережу, а достатньо просто зберегти вектор її ознак.

FaceNet продемонструвала блискучі результати на багатьох відомих тестових наборах. Наприклад, у задачах розпізнавання обличчя у відкритих та складних умовах вона перевершила попередні моделі, забезпечуючи надзвичайно високий рівень точності при порівняно низьких обчислювальних витратах [66]. У практиці це означає, що системи на базі FaceNet можуть працювати навіть на мобільних пристроях і в реальному часі.

Формальний опис цієї функції втрат описано згідно формули (3.2).

$$\|f(x^a) - f(x^p)\|^2 + \alpha < \|f(x^a) - f(x^n)\|^2, \quad (3.2)$$

де  $f(x^a)$  – вектор ознак, отриманий з якорного зображення;

$f(x^p)$  – вектор ознак, отриманий з позитивного прикладу;

$f(x^n)$  – вектор ознак, отриманий з негативного прикладу;

$\alpha$  – константа, що задає відступ між зображеннями різних людей.

Основна ідея використання цієї функції втрат полягає в тому, щоб навчити нейронну мережу генерувати вектори ознак, які для одного обличчя будуть дуже схожі (близькі за відстанню), а для різних осіб – відрізнятися на певну константу, яка називається відступом (рис. 3.5). Відступ необхідний, щоб уникнути тривіального рішення, коли мережа генерує однакові вектори для всіх людських образів.

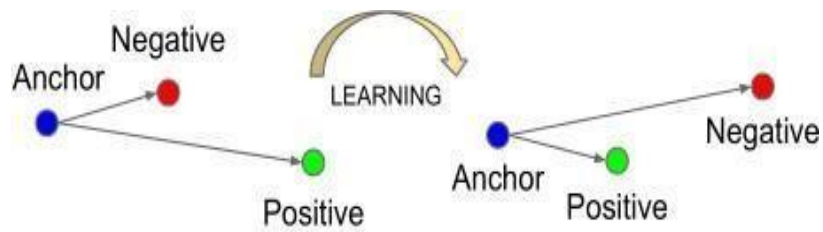


Рисунок 3.5 – Процес навчання при використанні Triplet Loss [66]

Важливим аспектом використання функції втрат Triplet Loss є те, що негативні приклади повинні бути вибрані середніми за складністю. Це означає, що зображення людей не повинні бути надто схожими, але й не дуже відрізнятися одне від одного. Дослідження показали, що така стратегія забезпечує найкращі результати навчання, оскільки мережа не потрапляє в локальні екстремуми під час градієнтного спуску.

Наступним етапом розвитку мережі FaceNet стала мережа OpenFace [67]. Її основна мета полягала в тому, щоб перевірити можливість навчання мережі для розпізнавання людських образів на відкритих наборах даних, на відміну від попередніх моделей, які використовували приватні датасети з мільйонами зображень. OpenFace також була орієнтована на роботу з мобільними пристроями з обмеженими обчислювальними потужностями, але мала бути достатньо швидкою для використання в реальному часі. Структура мережі OpenFace була модифікацією FaceNet з меншим числом параметрів для покращення швидкості навчання та роботи.

Як зазначалося раніше, для задач ідентифікації та верифікації часто застосовують метод кодування зображень людських образів у вектори ознак через навчання класифікаторів. Однак функція помилки Softmax не забезпечує чітке поділення класів, особливо на їх межах. Для вирішення цієї проблеми були запропоновані більш вдосконалені функції втрат, такі як ArcFace, яка була розроблена та протестована в однойменній нейронній мережі згідно формули (3.3).

$$L = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot w_{os}(0y_i) - m}}{e^{s \cdot w_{os}(0y_i) - m} + \sum_{j=1, j \neq y_i}^C e^{s \cdot w_{os}(0i)}}, \quad (3.3)$$

де  $L$  – значення функції втрат (loss), усереднене по всіх  $N$  прикладах;

$N$  – кількість прикладів у пакеті (batch size);

$C$  – кількість класів;

$i$  – індекс поточного прикладу в пакеті;

$y_i$  – правильний клас (мітка) для  $i$ -го прикладу.

Візуалізація прикладу класифікації за допомогою функції втрат ArcFace порівняно з Softmax наведена на рисунку 3.6.

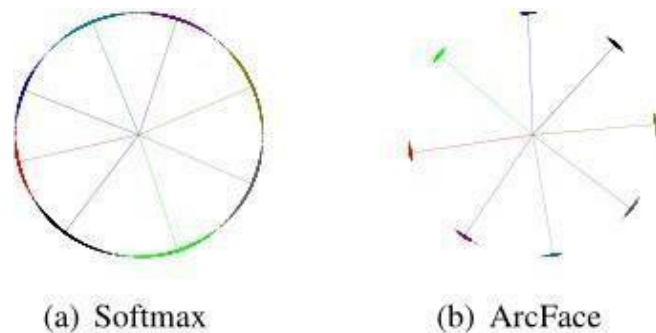


Рисунок 3.6 – Візуалізація класифікації зображень при використанні функцій помилок Softmax і ArcFace [67]

На зображенні різні кольори та лінії представляють різні класи. Можна помітити, що в Softmax межі між класами мають розмиті переходи, тоді як у ArcFace між класами є чіткі відстані, що ускладнює неправильне віднесення об'єкта до іншого класу.

### 3.2 Внесення змін до архітектури системи ідентифікації особи

Під час аналізу існуючих мереж було розглянуто багато різних архітектур. У випадку деяких з них основна увага зосереджувалася не стільки на самій архітектурі, скільки на підходах до їх навчання. Наприклад, для мереж, що перетворюють зображення людських образів в векторні подання, виділяють два основні методи.

Перший метод передбачає використання функції втрат Triplet Loss, яка сприяє зменшенню відстані між векторами ознак для зображень однієї особи та

збільшенню відстані між векторами для зображень різних людей. Другий метод полягає в навчанні класифікатора з подальшим видаленням останнього шару, що відповідає за класифікацію, в результаті чого утворюється мережа, здатна автоматично виділяти ознаки.

Для ефективного навчання з використанням Triplet Loss важливо підбирати триплети середньої складності, однак це є складним процесом, оскільки вручну потрібно вибирати правильні зображення для великого датасета, що є довготривалим і трудомістким завданням [68].

Структура мережі, описана в таблиці 3.1, включає в себе ці розгалуження, що утворюють final inception шари, що зображені на рисунку 3.7.

Таблиця 3.1 – Загальна структура мережі

№	Тип шару	Опис
1	вхідний шар	кольорове зображення розміром 96x96 пікселів
2	згортка	96 фільтрів розміром 7x7 пікселів з кроком 2x2
3	макс-пулінг	розмір 3x3 пікселів з кроком 2x2
4	згортка	64 фільтрів розміром 1x1 пікселів з кроком 1x1
5	згортка	192 фільтрів розміром 3x3 пікселів з кроком 1x1
6	макс-пулінг	розмір 3x3 пікселів з кроком 2x2
7	inception (3a)	конкатенує 4 відгалуження шарів з попереднього
8	inception (3b)	конкатенує 4 відгалуження шарів з попереднього
9	inception (3c)	конкатенує 3 відгалуження шарів з попереднього
10	inception (4a)	конкатенує 4 відгалуження шарів з попереднього
11	inception (4e)	конкатенує 3 відгалуження шарів з попереднього

Кінець таблиці 3.1

№	Тип шару	Опис
12	inception (5a)	конкатенує 3 відгалуження шарів з попереднього
13	inception (5b)	конкатенує 3 відгалуження шарів з попереднього
14	average-пулінг	розмір 3x3 пікселя з кроком 1x1

Мережа OpenFace, яка є розвитком FaceNet, використовує Triplet Loss і є однією з сучасних моделей для розпізнавання людських образів.

Як модифікація цієї мережі пропонується додавання класифікаційного шару в кінці OpenFace, що дозволить використовувати її для класифікації.

Крім того, для покращення швидкості навчання пропонується спростити структуру мережі, видаливши 4 та 5 групи inception шарів, що дозволить прискорити процес навчання без зниження якості розпізнавання. Відповідні зміни виділені на рисунку 3.7.

Також слід зазначити, що у всіх наведених таблицях та рисунках згортковий шар завжди складається з трьох компонентів: власне згортки, батч-нормалізації та активації за допомогою функції ReLU.

При додаванні класифікаційного шару до кінця цієї мережі вводиться новий повнозв'язний шар, у якому кількість нейронів відповідає числу класів, що використовуються під час навчання.

Кожен клас відображає окрему особу, а мережа навчається асоціювати зображення людських образів з відповідними класами.

Оскільки кількість класів залежить від розміру навчального набору даних, немає сенсу заздалегідь встановлювати фіксовану їх кількість.

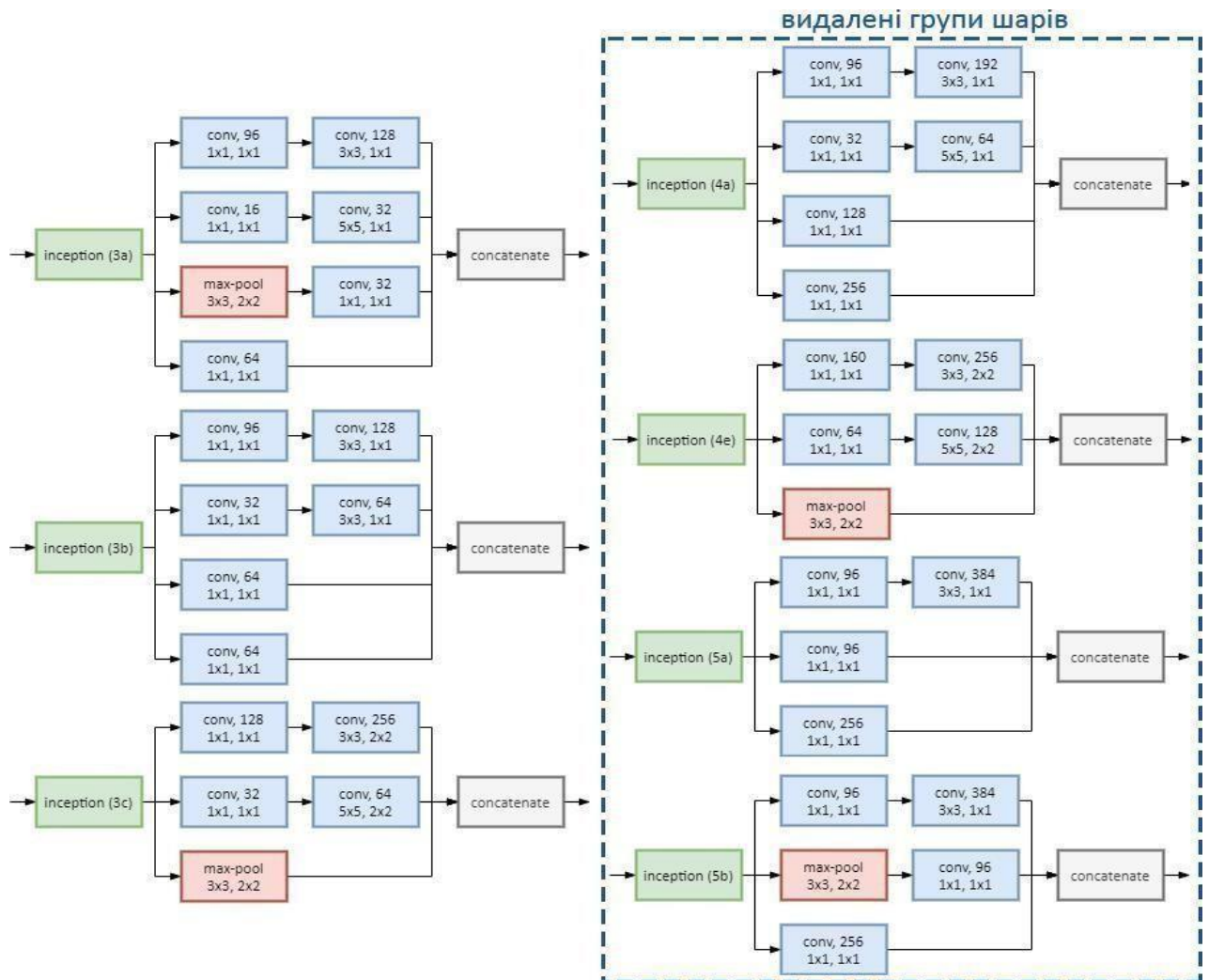


Рисунок 3.7 – Структура inception модулів оригінальної мережі OpenFace та видалена частина шарів в модифікованій версії

### 3.3 Висновки

У даному розділі було запропоновано власну модель системи ідентифікації особи з урахуванням впливу зовнішніх чинників, таких як освітлення, кут огляду та інші перешкоди. Модель використовує сучасні архітектури CNN і включає механізми оптимізації, що дозволяють зменшити кількість помилок. Також було оновлено структуру системи, інтегрувавши її у мультиагентне середовище для покращення швидкодії. Було проаналізовано можливі виклики при реальному застосуванні та запропоновано методи їх подолання.

## 4 РЕАЛІЗАЦІЯ ОПТИМІЗОВАНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ОСОБИ

### 4.1 Підготовка середовища для розробки

У сфері машинного навчання широко застосовується мова програмування Python. Це високорівнева кросплатформна мова, яка значно спрощує розробку завдяки великій кількості готових бібліотек. Серед них є інструменти для математичних обчислень, обробки зображень, а також бібліотеки, що полегшують створення та навчання складних нейронних мереж. Враховуючи всі ці переваги, для реалізації власного модуля штучної нейронної мережі було вирішено використовувати саме Python [69].

Оскільки навчання нейронних мереж вимагає виконання великої кількості обчислень, використання центральних процесорів (CPU) у таких завданнях значно поступається за швидкістю графічним процесорам (GPU). Саме тому в машинному навчанні переважно застосовують GPU разом із бібліотекою TensorFlow, яка дозволяє безпосередньо виконувати обчислення на графічному процесорі.

Створення складних архітектур нейронних мереж з нуля є досить трудомістким процесом, тому для глибокого навчання існує безліч спеціалізованих бібліотек, які спрощують цей процес. У проведених експериментах було обрано бібліотеку Keras, що має зручний інтерфейс і дозволяє конструювати мережі будь-якої складності з уже готових компонентів. Вона містить різноманітні шари, функції активації, методи оптимізації градієнтного спуску та багато інших інструментів, які значно полегшують створення нейромереж. Крім того, Keras дає можливість навчати модель і оцінювати якість її навчання на основі обчисленої похибки. Починаючи з версії 2.0, ця бібліотека включена до стандартного набору TensorFlow.

На додаток до основних бібліотек для машинного навчання – TensorFlow і Keras – було вирішено використовувати Numpy, Matplotlib і OpenCV. Бібліотека Numpy є ефективним інструментом для виконання математичних розрахунків, Matplotlib забезпечує побудову графіків, що корисно для аналізу результатів

навчання, а OpenCV надає широкий функціонал для роботи із зображеннями. Зокрема, OpenCV дозволяє завантажувати файли в різних популярних форматах, відображати їх, а також виконувати малювання, наприклад, наносити обмежувальні рамки на обличчя в зображеннях.

Для проведення експериментів із параметрами нейронної мережі зручним інструментом є інтерактивне середовище розробки Jupyter Notebook. Воно дозволяє виконувати код на Python у спеціальних блокнотах, де можна запускати окремі фрагменти коду та одразу переглядати їхні результати [70]. Всі отримані в ході виконання дані зберігаються, що дає змогу повторно їх переглядати, аналізувати або використовувати для подальших розрахунків. Зокрема, Jupyter Notebook спрощує візуалізацію даних, оскільки дозволяє безпосередньо виводити зображення та графіки без необхідності створення окремих віконних додатків.

Крім того, у Jupyter Notebook можна не лише писати код, а й додавати текстові описи у форматі Markdown. Це дає змогу використовувати можливості текстового редактора, змінювати розмір шрифту, застосовувати жирний або курсивний текст, змінювати колір та навіть використовувати елементи HTML-розмітки для більш гнучкого форматування. Така функціональність особливо корисна для документування експериментів, опису процесів чи постановки завдань безпосередньо під час роботи над кодом. Завдяки цьому Jupyter Notebook став популярним інструментом серед Python-розробників, науковців і дослідників, оскільки дозволяє швидко тестувати нові бібліотеки та методи, зберігаючи при цьому всі результати та пояснення.

Ще одним важливим аспектом роботи з Python є використання віртуального середовища. Воно забезпечує створення ізольованого оточення для проєктів, що містить власний інтерпретатор Python та набір бібліотек, необхідних для конкретного додатка. Це гарантує, що залежності будуть стабільними й не конфліктуватимуть між різними проєктами. Віртуальне середовище також спрощує перенесення програмного коду між різними машинами та забезпечує відтворюваність результатів, оскільки дозволяє використовувати ті самі версії бібліотек та інтерпретатора Python.

Таким чином, перед початком роботи було налаштовано віртуальне середовище та встановлено всі необхідні бібліотеки Python. Команди процесу налаштування середовища з використанням версії Python 3.13.2 прописуються наступним чином:

```
# Перехід у директорію проекту
Cd D: /Projects/face-recognition-cnn
# Створення та активація віртуального оточення у поточній директорії
Python -m venv.
scripts/activate
# Встановлення залежностей через Pip
pip install tensorflow = =2.4.1
pip install Keras = =2.4.3
pip install numpy = =1.19.5
pip install matplotlib = =3.4.0
pip install jupyterlab = =3.0.12
pip install opencv-python = =4.5.1.48
```

Бібліотека TensorFlow для роботи з графічними процесорами (GPU) потребує встановлення бібліотеки CUDA, яка використовується для обчислень на відеокартах NVIDIA. Тому для забезпечення коректної роботи було прийнято рішення використовувати комп'ютери, оснащені саме цими відеокартами. Щоб TensorFlow міг ефективно взаємодіяти з GPU, необхідно також встановити додаткові компоненти – CUDA Toolkit та cuDNN. Без цих бібліотек при виконанні програм, що використовують TensorFlow, можуть виникати помилки та попередження про відсутність необхідних DLL-файлів, що змушуватиме систему автоматично переходити на обчислення за допомогою CPU замість GPU.

Інсталятор CUDA Toolkit є безкоштовним і доступний для завантаження з офіційного сайту NVIDIA. Під час завантаження необхідно вказати операційну систему (ОС), її версію, а також вибрати відповідний тип інсталятора. У даному випадку було використано ОС Windows 10, яка підтримує архітектуру x64, що відповідає технічним вимогам робочої станції.

Існує два варіанти інсталятора: локальний і мережевий. Локальний інсталятор містить усі необхідні компоненти у файлі встановлення, тоді як

мережевий завантажує їх під час інсталяції [71]. Було обрано локальний варіант, оскільки він дозволяє уникнути проблем із встановленням у разі розриву інтернет-з'єднання. Процес встановлення CUDA детально представлений на рисунку 4.1.



Рисунок 4.1 – Процес інсталяції NVIDIA CUDA

Бібліотека cuDNN є доповненням до CUDA і працює безпосередньо разом із нею, проте її необхідно встановлювати окремо. Завантаження cuDNN було здійснено з офіційного сайту CUDA у форматі архіву, що містить файли C++ та DLL. Після цього ці файли потрібно було вручну розмістити у відповідних підкаталогах у директорії, де була встановлена CUDA [72].

Перед вибором та завантаженням відповідної версії cuDNN була проведена перевірка сумісності версій, використовуючи команду, наведену на рисунку 4.2.

TensorFlow активно використовує концепцію тензорів, які характеризуються певною розмірністю. Найчастіше застосовуються тензори розмірності від 0 до 3. Наприклад, 0D-тензор – це скалярне число, 1D-тензор являє собою вектор, 2D – це матриця, а 3D – тривимірний масив даних. Всі розрахунки з тензорами виконуються процесором, що може бути неефективним для великих обчислень.

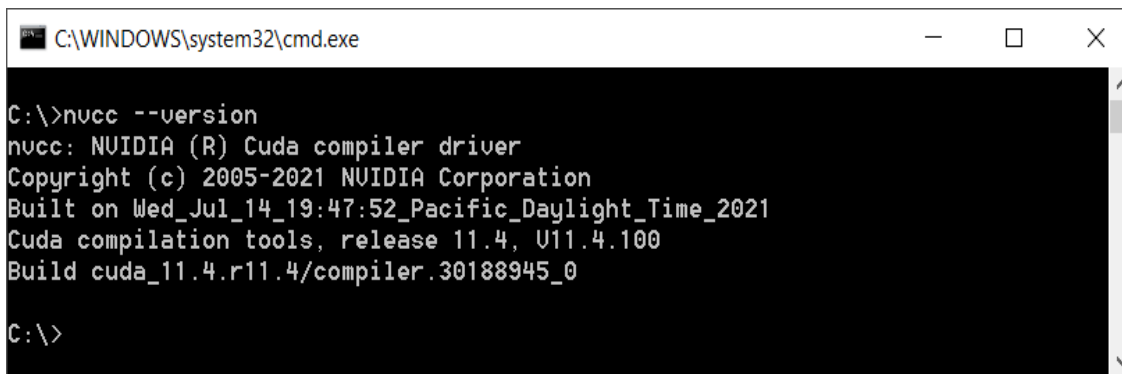
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the command 'C:\>nvc --version' and its output: 'nvc: NVIDIA (R) Cuda compiler driver', 'Copyright (c) 2005-2021 NVIDIA Corporation', 'Built on Wed\_Jul\_14\_19:47:52\_Pacific\_Daylight\_Time\_2021', 'Cuda compilation tools, release 11.4, U11.4.100', and 'Build cuda\_11.4.r11.4/compiler.30188945\_0'. The prompt returns to 'C:\>'.

Рисунок 4.2 – Перевірка версії встановленого CUDA

Щоб покращити швидкість і точність роботи з тензорами, компанія Google розробила власні спеціалізовані процесори – тензорні обчислювальні пристрої (TPU). Вони оптимізовані для задач машинного навчання і доступні для експериментів у хмарному середовищі Google Colab, де також можна використовувати і графічні процесори (GPU).

Google Colab [73] – це онлайн-платформа для обчислень, яка надає дослідникам безкоштовний доступ до ресурсів для розробки та навчання моделей машинного навчання. Вона має попередньо встановлені TensorFlow, Keras та інші популярні бібліотеки, що використовуються у сфері аналізу даних. Colab працює у форматі Jupyter Notebook прямо в браузері, дозволяючи виконувати код Python у зручному інтерактивному середовищі.

Середовище Google Colab працює на віртуальних машинах під керуванням операційної системи Linux, що дає змогу не тільки виконувати код Python, а й використовувати команди Bash. Це дозволяє переходити між каталогами файлової системи або встановлювати додаткові бібліотеки через менеджер пакетів.

Попри свої переваги, Colab має певні обмеження, зокрема щодо використання пам'яті та часу виконання Python-процесів. Однак Google не надає фіксованих обмежень, оскільки вони динамічно змінюються залежно від навантаження на сервери та кількості користувачів. Незважаючи на ці обмеження, Colab залишається зручним і потужним інструментом для тестування та навчання моделей нейронних мереж, дозволяючи працювати з сучасними

високопродуктивними апаратними ресурсами без необхідності встановлення додаткового програмного забезпечення на локальний комп'ютер.

Інтерфейс Google Colab із прикладами роботи можна побачити на рисунку 4.3.

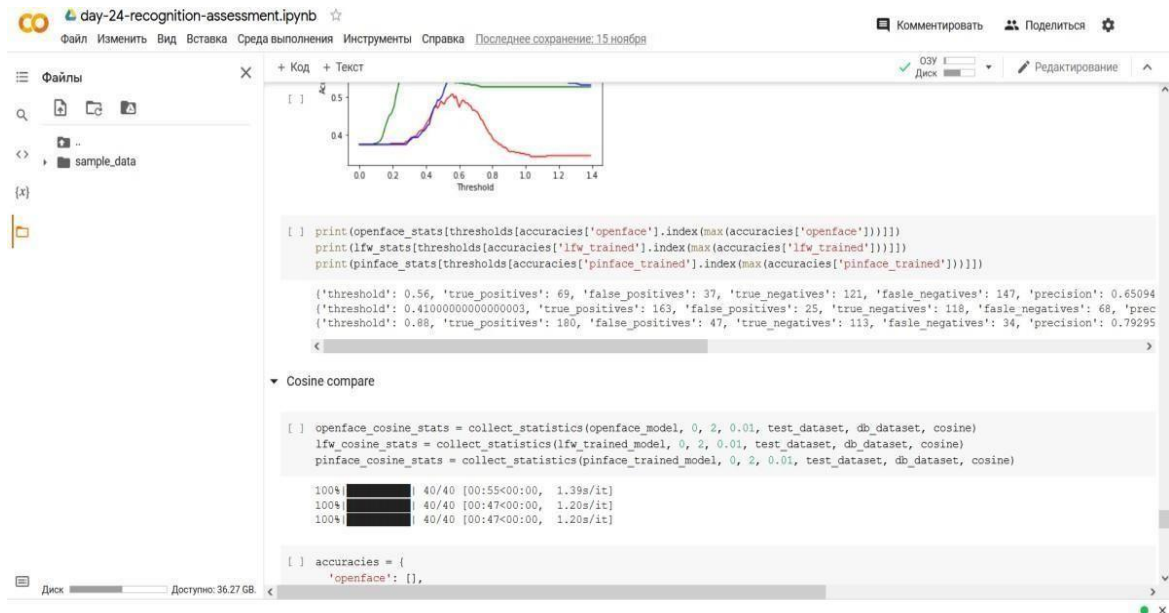


Рисунок 4.3 – Вигляд інтерфейсу Google Colab

Для написання кінцевого коду для використання нейронної мережі як середовища розробки було вибрано безкоштовний, легкий, але розширюваний редактор вихідного коду від Microsoft – Visual Studio Code. Він підтримує величезну кількість мов, в тому числі і Python у вигляді спеціального плагіна, який і був встановлений для подальшої роботи. Це дозволяє використовувати редактор як повноцінний IDE, який підтримує автодоповнення коду і налагодження.

## 4.2 Підготовка робочого середовища для розробки

Для ефективного навчання нейронних мереж, які використовуються для розпізнавання людських образів, необхідні великі набори зображень. Зазвичай такий набір розділяється на дві основні частини: навчальну та тестову. Навчальна

вибірка використовується для навчання моделі, а тестова – для перевірки її точності на нових, раніше невідомих для неї прикладах [74].

Окрім цього, часто з навчального набору виділяють ще одну підмножину – валідаційну вибірку. Вона необхідна для контролю якості навчання: мережу на ній не навчають, але після кожної епохи перевіряють рівень помилки. Якщо помилка на навчальних даних продовжує зменшуватися, а на валідаційній вибірці починає збільшуватися, це свідчить про перенавчання. У такому випадку варто припинити тренування, щоб уникнути погіршення узагальнюючої здатності моделі.

Створення власного набору зображень для навчання нейронної мережі – трудомісткий та витратний процес. Оскільки якісне навчання вимагає великого та збалансованого датасету, найчастіше використовують вже існуючі публічні набори даних. Деякі з них стали стандартами в галузі розпізнавання людських образів, дозволяючи об'єктивно порівнювати ефективність різних моделей.

Також певні датасети спеціально містять спотворені або перекриті зображення, щоб ускладнити завдання розпізнавання. Використання таких наборів допомагає моделям навчитися ідентифікувати людей за ключовими рисами обличчя, а не за другорядними деталями.

Серед численних публічних датасетів, які застосовуються в дослідженнях та розробці систем розпізнавання, було розглянуто та використано набір Labeled Faces in the Wild (LFW).

Набір Labeled Faces in the Wild (LFW). Цей датасет містить 13 233 зображення, що представляють 5 749 осіб. Він був створений на основі відкритих фотографій із мережі Інтернет, оброблених за допомогою детектора людських образів, реалізованого в OpenCV методом Віоли-Джонса [75].

Усі знімки є кольоровими та мають розмір 255×255 пікселів. Кожне зображення містить область голови людини, а кількість фотографій на одну особу може варіюватися від 1 до 530.

Файли представлені у форматі JPG та розподілені по 5749 директоріях, назви яких відповідають іменам та прізвищам людей у наборі, розділеним символом

підкреслення. У середині кожної директорії файли мають назву у форматі Ім'я\_Прізвище\_номер зображення.

Приклад зображень із датасету LFW представлено на рисунку 4.4.



Рисунок 4.4 – Приклади зображень із датасету LFW [15]

Набір PinsFace. Датасет PinsFace містить зображення відомих особистостей, зібрані з відкритих джерел в Інтернеті. Загалом у ньому налічується 17 534 зображення, які поділені між 105 осіб.

Приклад зображень із набору PinsFace представлено на рисунку 4.5.



Рисунок 4.5 – Приклади зображень із датасету PinsFace [15]

Усі знімки є кольоровими та демонструють обличчя в різних умовах: під різними кутами, з різними емоціями, при змінному освітленні, з макіяжем або без нього.

Структура датасету включає 105 директорій, кожна з яких містить понад 100 зображень, що відповідають певній людині. Назви директорій збігаються з іменами відомих особистостей, а файли всередині мають назву, що відповідає директорії, з доданим номером зображення.

Набір ORL Faces. Датасет ORL Faces містить зображення людських людських образів, отримані в контрольованих лабораторних умовах. Він включає 40 різних осіб, кожна з яких представлена десятьма фотографіями.

Для деяких людей фотографії робилися в різний час, змінюючи освітлення, вираз обличчя (наприклад, відкриті чи закриті очі, посмішка або її відсутність) і деталі зовнішності (наявність або відсутність окулярів).

Знімки зроблені на темному однорідному фоні, а випробувані перебували у фронтальному положенні, хоча допускалися незначні повороти голови [76]. Всі зображення є чорно-білими, мають формат PGM і роздільну здатність 92×112 пікселів.

Файли організовані в 40 папок, кожна з яких представляє окрему людину. Назви каталогів мають формат sX, де X – порядковий номер особи (від 1 до 40). Усередині кожної папки міститься десять зображень, імена яких мають вигляд Y.pgm, де Y – номер конкретного фото (від 1 до 10).

Приклад зображень із набору ORL Faces представлено на рисунку 4.6.

Вищенаведені набори зображень можна використовувати для навчання і тестування власної системи розпізнавання людських образів.

Для збільшення обсягу навчального набору зображень існує метод, відомий як аугментація даних (data augmentation). Його основна ідея полягає в створенні нових зображень на основі вже наявних шляхом застосування різних трансформацій та ефектів. Наприклад, можна виконати відображення зображення по вертикалі, змінити його розміри, додати розмиття або шуми. Також є можливість змінювати яскравість, контрастність чи колірний баланс зображення [77]. Такі

зміни можуть бути особливо корисними для покращення розпізнавання осіб при різному освітленні, що відрізняється від природного денного світла.



Рисунок 4.6 – Приклади зображень із датасету ORL Faces [18]

Зміщення зображення, наприклад, його переміщення на певну частину екрану, може сприяти покращенню розпізнавання навіть частин обличчя. Аугментація даних може бути реалізована безпосередньо в процесі навчання за допомогою різних бібліотек, що значно заощаджує час, оскільки не потрібно вручну виконувати ці трансформації.

#### 4.3 Алгоритми оцінювання продуктивності нейронної мережі

Для збільшення обсягу навчального набору зображень існує метод, відомий як аугментація даних (data augmentation). Його основна ідея полягає в створенні нових зображень на основі вже наявних шляхом застосування різних трансформацій та ефектів. Наприклад, можна виконати відображення зображення по вертикалі, змінити його розміри, додати розмиття або шуми. Також є можливість змінювати яскравість, контрастність чи колірний баланс зображення [78]. Такі

зміни можуть бути особливо корисними для покращення розпізнавання осіб при різному освітленні, що відрізняється від природного денного світла (таблиця 4.1).

Зміщення зображення, наприклад, його переміщення на певну частину екрану, може сприяти покращенню розпізнавання навіть частин обличчя. Аугментація даних може бути реалізована безпосередньо в процесі навчання за допомогою різних бібліотек, що значно заощаджує час, оскільки не потрібно вручну виконувати ці трансформації.

Таблиця .4.1 – Матриця заплутаності

Приклади	Позитивні	Негативні
Позитивні	Істинно позитивні(True Positive, TP)	Помилково позитивні (False Positive, FP)
Негативні	Помилково негативні (False Negative, FN)	Істинно негативні (True Negative, TN)

У контексті розпізнавання осіб, коли використовуються зображення з бази даних, що містить зображення осіб, матриця результатів класифікації буде інтерпретуватися таким чином:

- істинно позитивні приклади позначають ті зображення, які були правильно співвіднесені з відповідними людьми з бази даних, тобто класифікація є вірною;
- помилково позитивні приклади позначають зображення, які були неправильно співвіднесені з людьми з бази даних, хоча насправді ці зображення не належать їм, тобто класифікація помилкова;
- істинно негативні приклади позначають зображення, які не були співвіднесені з жодною особою з бази даних, і насправді вони не належать жодній людині з бази даних, тобто класифікація вірна;
- помилково негативні приклади позначають зображення, які не були співвіднесені з жодною особою з бази даних, але насправді вони належать комусь з цієї бази, тобто класифікація помилкова

Перед оцінкою якості розпізнавання тестовий набір даних поділяється на дві частини: одна частина зображень використовується для співвіднесення з базою даних, а інша частина перевіряється на відповідність чи невідповідність жодній людині з бази.

Після цього підраховується кількість зображень, що потрапили в одну з груп матриці заплутаності. На основі цих результатів обчислюються метрики, що дають більш загальне уявлення про точність розпізнавання. Однією з основних та найпоширеніших метрик є точність (*precision*), яка вказує на частку правильно виявлених зображень. Вона також відома як надійність або повторюваність, що означає, що при повторних вимірюваннях в однакових умовах будуть отримані рівні результати. У бінарній класифікації точність іноді називають позитивним прогностичним значенням. Для розрахунку точності застосовують формулу (4.1) .

$$Precision = \frac{TP}{TP+FP}, \quad (4.1)$$

де TP – істинно-позитивні спрацювання;

FP – хибно-позитивні спрацювання.

Чуйність (*recall*) – це відсоток позитивних випадків, які були вірно ідентифіковані. Вона показує, яку частину релевантних випадків система змогла правильно виявити.

Після визначення метрик для оцінки навчених моделей і їх порівняння, потрібно вибрати відповідний тестовий набір даних для цього процесу. Оскільки кожне зображення людей необхідно порівнювати з усіма іншими зображеннями, кількість операцій значно збільшується. Тому було прийнято рішення вибрати компактний набір даних для тестування й оцінки моделей. Для цього був обраний набір ORL Faces, який складається з 40 різних осіб, кожен з яких має 10 зображень. Фотографії зроблені в різних умовах – з різним освітленням, у різний час доби, з різними виразами обличчя та аксесуарами (окуляри, наприклад) [79]. Оскільки розмір цього набору невеликий, його не слід використовувати для навчання мережі,

якщо потрібно досягти високих результатів, але він ідеально підходить для тестування.

Тому було вирішено випадковим чином розподілити цей датасет на дві частини: базу даних і тестову вибірку, де ймовірність того, що зображення потрапить у базу даних, складає 50%. У базу даних потрапляє лише одне зображення кожної особи, решта зображень стають частиною тестового набору. Таким чином, у базі даних для кожної людини є тільки одне зображення, і мережа повинна співвіднести або не співвіднести кожне зображення з тестової вибірки з відповідним обличчям у базі.

Після обчислення значень метрик схожості між векторами необхідно прийняти рішення щодо того, чи є відстань між ними достатньо малою, щоб вважати їх однаковими для однієї людини. Для цього вводиться додатковий параметр алгоритму порівняння – поріг. Цей параметр потрібно підбирати для кожної моделі нейронної мережі шляхом експериментів. Метою такого підбору є знаходження значення порогу, яке забезпечить найкращі результати для конкретної моделі. Порівняння між векторами здійснюється за допомогою евклідової відстані, але при необхідності можна застосовувати й інші методи порівняння векторів.

#### 4.4 Створення класу для процесу навчання та тестування

Структура програмного забезпечення для розробки, навчання та оцінки нейронної мережі, призначеної для розпізнавання людських образів, відображена на класових діаграмах, зображених на рисунках 4.7 та 4.8. На першому рисунку показана архітектура основних класів, що використовуються в процесі навчання, а на другому – архітектура компонентів, пов'язаних з алгоритмами навчання та оцінки мережі.

Один з основних класів – це клас FaceRecognitionNet. Він є абстрактним, оскільки його завданням є надання інтерфейсу для мережі розпізнавання людських образів, який потім буде успадковано іншими класами. Це дозволяє спростити реалізацію, виділивши загальні функції. Такі спадкоємці включають реалізацію

оригінальної мережі OpenFace та її модифіковану версію. Завдяки поліморфізму можна використовувати будь-яку реалізацію класу мережі для розпізнавання без необхідності змінювати основний алгоритм.

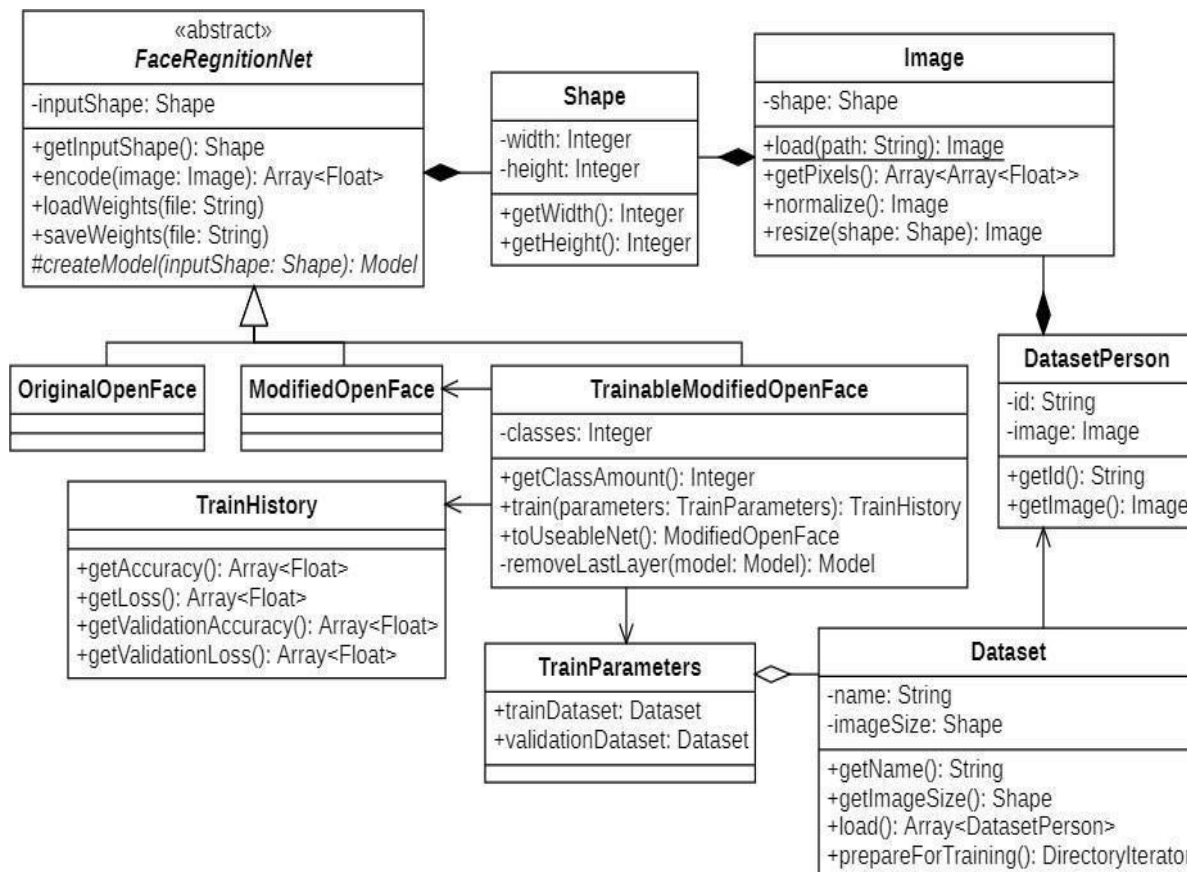


Рисунок 4.7 – Діаграма загальних класів та класів для реалізації навчання модифікованої мережі розпізнавання людських образів

Клас FaceRecognitionNet має наступні поля та методи:

- поле `inputShape` зберігає розміри зображення, яке подається на вхід мережі. Це значення може бути фіксованим у спадкоємці або передаватися як параметр через конструктор чи сеттер, тобто бути динамічним;
- метод `getInputShape` повертає розмір зображення, встановлений у полі `inputShape`;
- метод `encode` приймає зображення, що є фотографією обличчя людини, і кодує його в вектор ознак, який повертається у вигляді масиву чисел;

- метод `loadWeights` отримує шлях до файлу та завантажує ваги в модель нейронної мережі, замінюючи поточні;
- метод `saveWeights` отримує шлях до файлу та зберігає поточні ваги моделі;
- абстрактний метод `createModel` приймає розмір зображення, що використовуватиметься в моделі, будує модель і повертає її як екземпляр класу `Model` з бібліотеки Keras. Цей метод використовується виключно всередині класу `FaceRecognitionNet` для внутрішніх операцій.

Діаграма класів для алгоритмів розпізнавання та оцінки ефективності мережі зображена на рисунку 4.8.

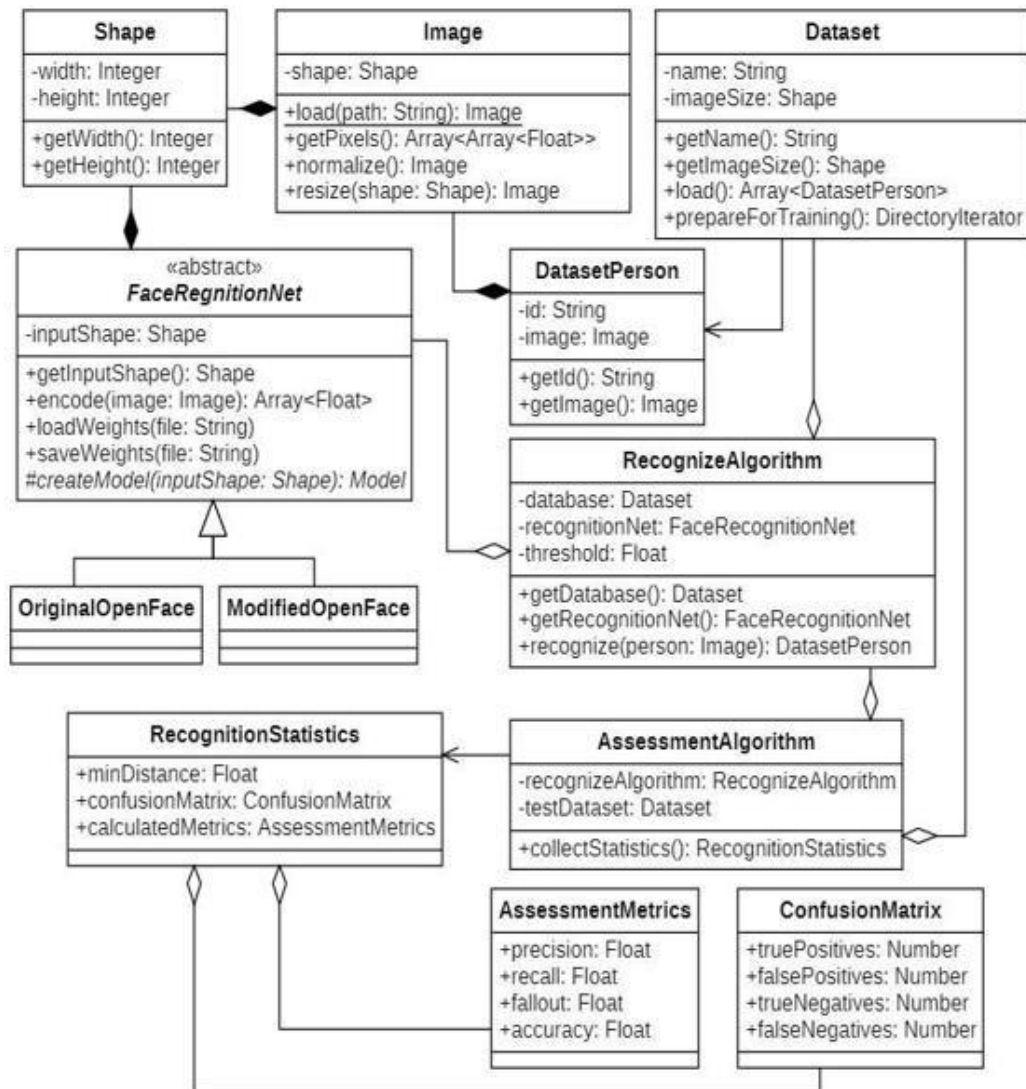


Рисунок 4.8 – Діаграма класів для реалізації алгоритмів розпізнавання людських образів та оцінювання нейронної мережі

Для забезпечення зручнішого доступу до елементів датасету, зокрема при розпізнаванні, використовується клас `DatasetPerson`.

#### 4.5 Програмна реалізація та тренування системи розпізнавання обличчя

Для розробки нейронної мережі для розпізнавання людських образів основний етап програмування був виконаний у Google Colab, оскільки цей сервіс вже має необхідні бібліотеки для машинного навчання. Після написання реалізації та проведення кількох тестових експериментів для перевірки коректності виконання коду Python, основне навчання було продовжено на власному комп'ютері.

Навчання проводилось двічі на різних датасетах: `PinFaces` і `LFW` [80].

Процес навчання тривав два дні. Перший день був присвячений навчанні на датасеті `PinFaces`, який містить 105 класів, а другий – на датасеті `LFW` з 5749 класами. Хоча кількість класів у цих датасетах сильно відрізняється, їх загальна кількість зображень була приблизно однаковою, тому час навчання на кожному з них був подібним. Крім того, кожен з датасетів був розширений завдяки аугментації даних за допомогою `Keras`. Зображення випадковим чином переміщувались на 10% в бік, обертались на 40 градусів, масштабувались на 20% і дзеркально відображались по вертикалі. Це все здійснювалося автоматично за допомогою завантажувача даних `Keras`, який поступово завантажує зображення з диска, оскільки великі датасети не завжди поміщаються в оперативній пам'яті, а також виконує оптимізації для прискорення навчання в `TensorFlow`.

Під час навчання на кожному датасеті зображення випадковим чином ділилися на навчальну і валідаційну вибірки: 80% зображень йшло на навчання, а 20% – на валідацію. Завдяки вбудованим функціям `Keras` було налаштовано автоматичне збереження ваг моделі при досягненні найкращих показників точності на валідаційній вибірці після кожної епохи. Якщо точність не покращувалася або погіршувалася, нові ваги не зберігалися.

Навчання припинялося, якщо протягом кількох епох точність не змінювалася. За замовчуванням, навчання проводилося до 500 епох. Якщо точність все ще покращувалася, навчання продовжувалося ще на 500 епох, поки не досягався оптимальний результат. Усього на датасеті LFW навчання тривало 794 епохи, а на PinFaces – 906, з досягнутими точностями на валідаційних вибірках 97% і 98% відповідно.

На рисунку 4.9 показано, як змінювалася втрата на валідаційній вибірці для датасету LFW. Як видно, помилка зменшувалась стабільно протягом усього навчання, і в кінці мережа досягла рівня помилки близько 0,25%.

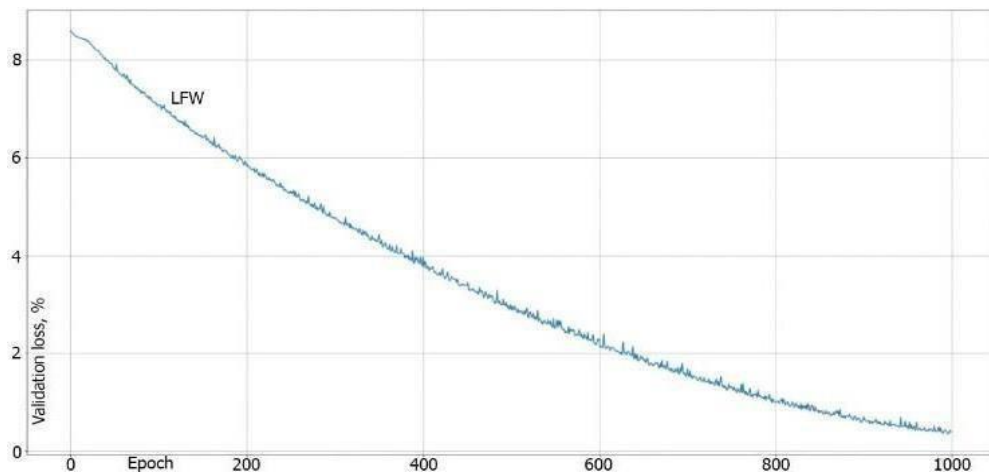


Рисунок 4.9 – Валідація на датасеті LFW

Динаміка зміни помилки на валідаційній вибірці для датасету PinFaces показана на рисунку 4.10.

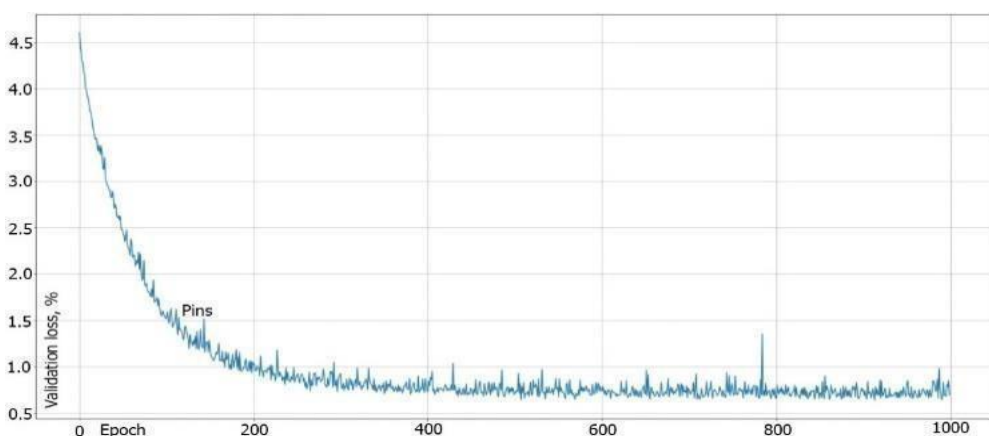


Рисунок 4.10 – Валідація на датасеті PinsFace

Оскільки PinFaces містить значно менше класів порівняно з LFW, на графіку можна побачити, що мережа досягла низького рівня помилки швидше – приблизно після 300 епох, і надалі помилка залишалася стабільною.

Аналогічні висновки можна зробити, розглядаючи графіки зміни точності класифікації для обраних датасетів, показані на рисунку 4.11.

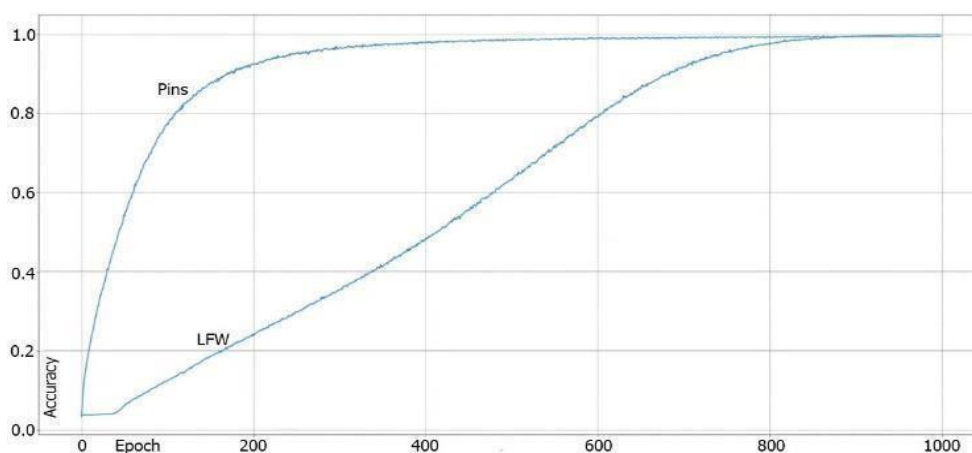


Рисунок 4.11 –Точність навчання на використаних датасетах

Під час навчання на PinFaces мережа досягає точності, близької до 100%, що майже вдвічі швидше, ніж при навчанні на LFW. Це дозволяє зробити висновок, що для досягнення оптимальних результатів на PinFaces потрібно близько 500 епох, тоді як для LFW достатньо 850-900 епох.

#### 4.6 Аналіз ефективності розробленої системи

Під час оцінювання, крім двох навчальних моделей, також була перевірена оригінальна попередньо навчена мережа OpenFace. Її вихідний код було знайдено на GitHub. Варто зазначити, що спочатку ця мережа була створена за допомогою бібліотеки Pytorch, яка є аналогом TensorFlow і також використовується для машинного навчання. Оскільки Pytorch несумісний з Keras, на якій здійснювалася розробка і навчання власної мережі, було використано її версію, конвертовану в Keras з тими ж вагами.

Що стосується підбору порогового значення під час оцінки мереж, то цей процес можна здійснити вручну, однак це займає багато часу, і можна випадково пропустити оптимальне значення. Тому було вирішено реалізувати пошук порогу шляхом грубого перебору в циклі. Першочергові експерименти допомогли визначити орієнтовний діапазон, в якому варто шукати поріг. Виявилось, що значення порогу може бути як меншим за нуль, так і більшим за одиницю, але не більше двох. Отже, було вибрано діапазон  $[0; 2)$  з кроком 0.01 для пошуку найкращих значень порогу.

Після проведення експериментів і налаштування параметрів алгоритму порівняння для кожної з моделей, були отримані результати, які показано нижче. На рисунку 4.12 представлено графік, який демонструє залежність точності розпізнавання від порогового значення для кожної з мереж, отриману при порівнянні векторів ознак через Евклідову відстань.

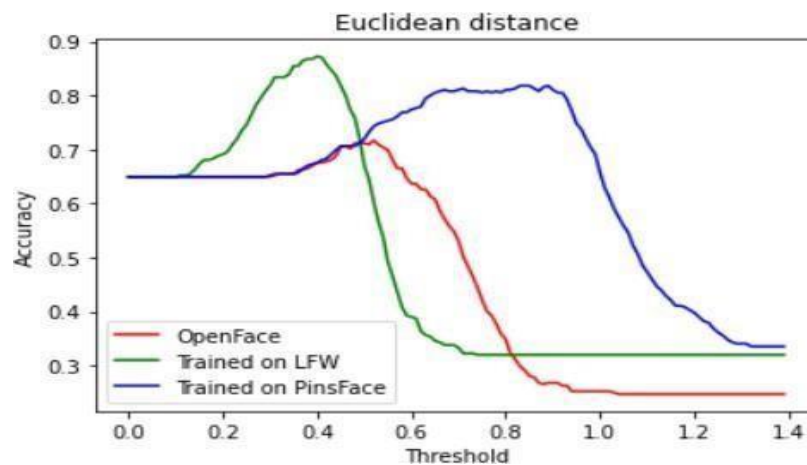


Рисунок 4.12 – Залежність акуратності розпізнавання від порога порівняння векторів по Евклідовій відстані

З графіка можна чітко побачити, що для модифікованої мережі точність розпізнавання при оптимальному порозі вища, ніж для оригінальної. Різниця між двома навченими мережами невелика, хоча варіації графіків помітні. Графік для мережі, навченої на датасеті LFW, має вужчу форму, тоді як графік для мережі, навченої на PinFaces, ширший. Це свідчить про те, що якість розпізнавання менш залежить від порогу в разі зменшеної кількості класів на датасеті PinFaces. З цього

можна зробити висновок, що залежність точності від порогу корелює з числом класів: чим менше класів, тим меншою є ця залежність.

Для кожної мережі було розраховано повний набір метрик у точці, що відповідає максимальній точності. Отримані результати представлені в таблиці 4.2. Зліва показані результати для оригінальної мережі OpenFace, посередині – для модифікованої мережі, навченої на датасеті LFW, а справа – для модифікованої мережі, навченої на датасеті PinFaces.

У першому рядку таблиці для кожної моделі вказано порогове значення, при якому вектори вважаються такими, що належать одній і тій самій особі. Це значення різне для кожної моделі, оскільки кожна модель була навчена за різними методами та/або на різних датасетах, тому поріг потрібно визначати для кожної окремо. Далі надані значення категорій матриці заплутаності, а також результати метрик точності, чутливості, специфічності та акуратності.

Як показали результати навчання власних моделей, вдалося збільшити загальну точність розпізнавання на 10-16%. Мережа, яка навчалася на датасеті LFW, продемонструвала кращий результат на 6% порівняно з мережею, яка працювала на датасеті PinFaces. Це, ймовірно, зумовлено більшою кількістю класів у LFW (5749) порівняно з 105 класами в PinFaces, що дозволяє моделі краще відрізняти більш схожі між собою зображення людських образів.

Таблиця 4.2 – Порівняння найкращих результатів моделей при порівнянні векторів за допомогою Евклідової відстані

Параметри	Оригіналь на OpenFace	Навчена на LFW	Навчена на PinFaces
Поріг	0.52	0.4	0.83
Число TP прикладів	34	104	102
Число FP прикладів	10	21	41

Кінець таблиці 4.2

Параметри	Оригіналь на OpenFace	Навчена на LFW	Навчена на PinFaces
Число TN прикладів	242	232	213
Число FN прикладів	99	28	29
Точність	77.27%	83.2%	71.33%
Чуйність	25.56%	78.79%	77.86%
Випадання	96.03%	91.7%	83.86%
Акуратність	71.69%	87.27%	81.81%

Ці висновки узгоджуються з дослідженням, описаним , де також підкреслюється вплив кількості класів на точність розпізнавання. Окрім цього, аналіз значень TP, FP, TN і FN показав, що оригінальна мережа, навіть при найвищій точності, більше орієнтується на негативні результати, відносячи обличчя до категорій тих, хто не представлений у базі даних, що інколи відбувається неправильно. Водночас модифіковані мережі продемонстрували кращу збалансованість у розподілі позитивних і негативних результатів, що призвело до вищої загальної точності.

Аналогічно було проведено збір і аналіз статистики при порівнянні векторів ознак за допомогою косинусної подібності. Метод грубого перебору порогових значень у циклі дозволив знайти оптимальні параметри, після чого було розраховано інші метрики. Графік залежності точності від порогу, наведений на рисунку 4.13, показав, що результати для двох методів порівняння векторів ознак майже однакові, із незначними відмінностями в деталях, таких як форма ліній на графіку.

Отже, можна зробити висновок, що для задачі розпізнавання осіб вибір методу порівняння векторів ознак не є критичним – обидва методи є придатними

для цього завдання. Відповідні метрики для найкращих значень порога, отриманих за допомогою косинусної подібності, подано в таблиці 4.3.

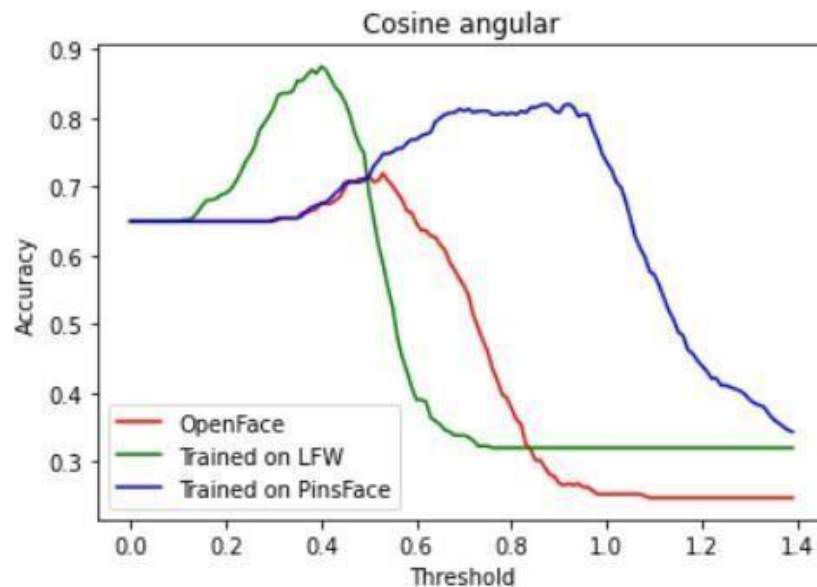


Рисунок 4.13 – Залежність акуратності розпізнавання від порога при порівнянні векторів по куту косинуса

Таблиця 4.3 – Порівняння найкращих результатів моделей при порівнянні векторів за допомогою косинуса кута

Параметри	Оригінальна OpenFace	Навчена на LFW	Навчена на PinFaces
Поріг	0.53	0.4	0.87
Число TP прикладів	35	103	106
Число FP прикладів	10	19	44
Число TN прикладів	242	234	210
Число FN прикладів	98	29	25
Точність	77.77%	84.43%	70.66%
Чуйність	26.32%	78.03%	80.92%

Кінець таблиці 4.3

Параметри	Оригінальна OpenFace	Навчена на LFW	Навчена на PinFaces
Випадання	96.03%	92.5%	82.68%
Акуратність	71.95%	87.53%	82.08%

Під час тестування також було виміряно час розпізнавання для кожної з мереж. Результати порівняння представлені на графіку, зображеному на рисунку 4.14. Значення, які наведено, відображають середній час, необхідний для розпізнавання одного тестового зображення серед усіх зображень у базі даних, тобто для ідентифікації однієї особи. Як видно, оригінальна мережа потребує близько 1,5 секунди, в той час як у модифікованій мережі вдалося зменшити цей час приблизно на 0,25 секунди.



Рисунок 4.14 – Середній час розпізнавання на одну ітерацію

Окрім оцінки точності розпізнавання, було проведено експеримент для порівняння швидкості навчання на різних процесорах. За результатами навчання мереж на TensorFlow (рис. 4.15), час, який витрачається на всі етапи навчання, залишається приблизно однаковим. Тому, щоб уникнути перенавчання мережі, було вирішено провести навчання модифікованої мережі на 100 епохах як на CPU,

так і на GPU. Навчання виконувалося на датасеті LFW, який демонструє кращі результати розпізнавання. У підсумку час навчання на CPU склав 23146 секунд, а на GPU – 4606 секунд, що відповідає приблизно 6,4 години та 1,3 години відповідно. Також була здійснена спроба навчання на сервісі Google Colab за допомогою процесорів TPU. Оскільки цей сервіс безкоштовний і одночасно використовується великою кількістю людей, він призначений для інтерактивної роботи. Тому в процесі використання сервісу часто виникають перерви через необхідність підтвердження присутності, що ускладнює тривале навчання на цьому ресурсі.

```

Epoch 1/100
410/410 [=====] - 405s 982ms/step - loss: 8.6107 - accuracy: 0.0381 - val_loss: 8.5949 - val_accuracy: 0.0025
Epoch 2/100
410/410 [=====] - 212s 518ms/step - loss: 8.4912 - accuracy: 0.0384 - val_loss: 8.5653 - val_accuracy: 0.0025
Epoch 3/100
410/410 [=====] - 213s 520ms/step - loss: 8.4243 - accuracy: 0.0384 - val_loss: 8.5447 - val_accuracy: 0.0025
Epoch 4/100
410/410 [=====] - 214s 522ms/step - loss: 8.3643 - accuracy: 0.0384 - val_loss: 8.5265 - val_accuracy: 0.0025
Epoch 5/100
410/410 [=====] - 216s 526ms/step - loss: 8.3074 - accuracy: 0.0384 - val_loss: 8.5104 - val_accuracy: 0.0025
Epoch 6/100
410/410 [=====] - 215s 524ms/step - loss: 8.2533 - accuracy: 0.0384 - val_loss: 8.4959 - val_accuracy: 0.0025
Epoch 7/100
410/410 [=====] - 214s 523ms/step - loss: 8.2018 - accuracy: 0.0384 - val_loss: 8.4838 - val_accuracy: 0.0025
Epoch 8/100
410/410 [=====] - 216s 528ms/step - loss: 8.1534 - accuracy: 0.0384 - val_loss: 8.4729 - val_accuracy: 0.0025
Epoch 9/100
410/410 [=====] - 217s 529ms/step - loss: 8.1080 - accuracy: 0.0384 - val_loss: 8.4634 - val_accuracy: 0.0025
Epoch 10/100
410/410 [=====] - 218s 532ms/step - loss: 8.0661 - accuracy: 0.0384 - val_loss: 8.4580 - val_accuracy: 0.0025
Epoch 11/100
410/410 [=====] - 218s 533ms/step - loss: 8.0274 - accuracy: 0.0385 - val_loss: 8.4513 - val_accuracy: 0.0027
Epoch 12/100
410/410 [=====] - 220s 536ms/step - loss: 7.9927 - accuracy: 0.0385 - val_loss: 8.4463 - val_accuracy: 0.0030

```

Рисунок 4.15 – Процес навчання мереж на Tensorflow з Keras

Зважаючи на це, експерименти на Google Colab були проведені лише на 10 епохах, з подальшим розрахунком часу, який був би необхідний для навчання на 100 епохах. Для розрахунку часу решти епох було враховано лише час етапів, що слідує після першої, оскільки, як показано на рисунку 4.15, перша епоха вимагає найбільше часу, а наступні етапи займають значно менше часу завдяки кешуванню даних під час першої епохи. Таким чином, час навчання на Google Colab для CPU склав близько 46900 секунд, на GPU – 8285 секунд, а на TPU – 47778 секунд, що

відповідає приблизно 13 годинам, 2,3 годинам та 13,3 годинам відповідно. Підсумкові результати дослідження швидкості навчання на CPU та GPU робочої станції кафедри ПМІ, а також на CPU, GPU та TPU сервісу Google Colab наведено у таблиці 4.4.

На основі результатів вимірювання швидкості навчання можна зробити висновок, що використання GPU на робочій станції кафедри ПМІ дозволяє суттєво скоротити час тренування моделі – у кілька разів у порівнянні з CPU.

Таблиця 4.4 – Приблизний час навчання на робочій станції ПМІ та виділеній машині Google Colab на CPU, GPU та TPU

Тип машини	Робоча станція	Виділена машина Google Colab
CPU	6,4 годин	13 годин
GPU	1,3 годин	2,3 годин
TPU	1,0 годин	1,7 годин

Крім того, під час роботи на Google Colab було виявлено, що TPU демонструє значно нижчу продуктивність порівняно з традиційним GPU і навіть поступається CPU за швидкістю. Отже, при проведенні експериментів з глибоким навчанням на платформі Google Colab доцільніше обирати GPU як оптимальний варіант для тренування нейронних мереж.А

#### 4.7 Висновки

У даному розділі представлено практичну реалізацію програмної частини розробленої системи. Проведено підготовку середовища розробки, створено класи для навчання та тестування моделі, реалізовано алгоритми оцінки продуктивності. Було проведено навчання системи на спеціалізованій вибірці та здійснено тестування з метою перевірки точності, швидкості та стійкості до зовнішніх чинників. За результатами експериментів встановлено, що запропонована система перевершує аналогічні рішення за більшістю ключових параметрів.

## ВИСНОВКИ

У процесі виконання магістерської кваліфікаційної роботи першочергово було здійснено глибокий аналіз предметної області, пов'язаної з автоматичним виявленням та розпізнаванням людських образів. В результаті цього аналізу були виявлені основні труднощі та обмеження, що виникають при побудові подібних систем. Зокрема, було розглянуто різні підходи до розпізнавання людських образів, а також їх переваги й недоліки. Зіставлення сучасних методів засвідчило, що використання нейронних мереж є одним із найбільш ефективних рішень завдяки високій точності, гнучкості та здатності до навчання на різноманітних даних.

У першому розділі проведено аналіз сучасного стану технологій ідентифікації особи, особливу увагу приділено нейронним мережам як основі для розпізнавання обличчя. Було розглянуто переваги біометричних методів порівняно з традиційними, а також проаналізовано найефективніші програмні засоби, такі як Azure Face API, Face ID і Cloud Vision API. Встановлено, що найбільш перспективним напрямком є застосування згорткових нейронних мереж для ідентифікації осіб у реальному часі.

У другому розділі обґрунтовано вибір нейронних мереж як основного інструменту для побудови системи розпізнавання обличчя. Розглянуто архітектуру системи, визначено структуру та особливості згорткових і повнозв'язних мереж, активаційні функції, алгоритми навчання та оптимізації. Запропоновано концептуальну модель системи, що враховує основні параметри зображень, які впливають на точність розпізнавання (освітлення, кут огляду тощо).

У третьому розділі розроблено архітектуру та математичну модель системи ідентифікації особи. Визначено критерії оптимізації — точність, швидкість і надійність. Запропоновано використання мультиагентної системи для інтеграції з нейронними мережами, що дозволяє підвищити ефективність функціонування системи в реальних умовах.

У четвертому розділі реалізовано програмну частину розробленої системи, підготовлено середовище для навчання нейромережі, розроблено класи для

тренування та тестування, а також реалізовано алгоритми оцінювання продуктивності. Проведено експериментальні дослідження, які підтвердили, що запропонована система забезпечує вищу точність та стійкість до змін зовнішніх умов порівняно з іншими методами. Результати експериментів засвідчили переваги обраного підходу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мартинюк Б. О., Грига В. М. Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі. *Матеріали XII Всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих вчених з автоматичного управління присвячені Дню ракетно-космічної галузі України: збірник матеріалів конференції (10-12 квітня 2025 р., м. Херсон, м. Хмельницький)*. Херсон, 2025. С. 85–87.
2. Герасимчук В. І., Ковальов О. М. Основи комп'ютерного зору : навч. посіб. Київ : Ліра-К, 2020. 212 с.
3. Шкарупило В. В., Місюра М. Д. Біометричні системи автентифікації : навч. посіб. Харків : ХНУРЕ, 2021. 228 с.
4. Сіренко І. М. Інтелектуальні системи розпізнавання образів : монографія. Львів : ЛНУ, 2021. 298 с.
5. Чалий О. В., Герасименко А. П. Машинне навчання : підручник. Харків : Факт, 2019. 256 с.
6. Козаченко Т. П. Основи нейронних мереж : навч. посіб. Київ : Академперіодика, 2020. 176 с.
7. Гудфеллоу І., Бенджіо Й., Курвіль А. Глибоке навчання. Київ : Видавництво Ліра-К, 2022. 774 с.
8. Білоус В. М., Кравченко Т. А. Інформаційні системи безпеки : навч. посіб. Харків : Університет внутрішніх справ, 2019. 312 с.
9. Мельник П. І. Штучний інтелект : навч. посіб. Київ : КНЕУ, 2021. 198 с.
10. Савченко О. І. Математичні методи в інформатиці : навч. посіб. Львів : Видавництво ЛНУ, 2020. 240 с.
11. Боднарчук Т. М. Машинне навчання та розпізнавання образів : монографія. Тернопіль : ТНТУ, 2019. 215 с.
12. Bishop C. M. *Pattern Recognition and Machine Learning*. New York : Springer, 2006. 738 p.
13. Aggarwal C. C. *Neural Networks and Deep Learning*. Cham : Springer, 2018.

497 p.

14. Zhang Y. Deep Learning for Face Recognition. Singapore : Springer, 2021. 245 p.
15. Gonzalez R. C., Woods R. E. Digital Image Processing. Boston : Pearson, 2018. 1136 p.
16. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521. P. 436–444.
17. Taigman Y., Yang M., Ranzato M. A., Wolf L. DeepFace: Closing the gap to human-level performance in face verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition: Columbus, USA, 2014*, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.
18. Schroff F., Kalenichenko D., Philbin J. FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
19. Алієв З. Системи біометричної ідентифікації: нейромережеві підходи : монографія. Київ : Наук. думка, 2023. 284 с.
20. Гуменюк І., Синиця О. Глибокі згорткові мережі у розпізнаванні образів. *Комп'ютерні інженерні технології*. 2024. № 2. С. 45–57.
21. Жовнерчук Т. Оптимізація процесу навчання нейронних мереж методом адаптивного моменту : підручник. Львів : ЛНУ ім. І. Франка, 2024. 176 с.
22. Dong C., Loy C. C., He K., Tang X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016. Vol. 38. №2. P. 295–307, doi: 10.1109/TPAMI.2015.2439281.
23. Коваленко М., Литвин В. Методи оцінювання точності біометричних систем на базі штучного інтелекту. *Інформаційні технології та безпека*. 2025. Т. 11, № 1. С. 12–26.
24. Мартиненко А. С. Нейромережеві технології у безпекових системах : монографія. Одеса : ОНАХТ, 2020. 165 с.
25. Степаненко І. В. Комп'ютерна графіка та обробка зображень : навч.

посіб. Київ : ВЦ «Академія», 2021. 184 с.

26. Яковенко С. П. Обробка цифрових зображень : навч. посіб. Харків : ХНУРЕ, 2021. 152 с.

27. Соловійов В. М., Іванов О. Л. Бази даних у системах безпеки. Львів : Новий Світ, 2021. 160 с.

28. Саченко Ю. О., Білик І. Б. Мережеві технології : навч. посіб. Тернопіль : ТНТУ, 2020. 268 с.

29. Могильний О. М. Інформаційна безпека : навч. посіб. Київ : КНЕУ, 2020. 210 с.

30. Кондратенко Ю. П. Методи захисту інформації в комп'ютерних системах : підручник. Одеса : ОНПУ, 2020. 234 с.

31. Дуброва Т. О. Штучний інтелект у практиці розпізнавання образів : монографія. Київ : НАУ, 2020. 238 с.

32. Гнатюк С. Н. Біометричні системи : теорія і практика. Львів : ЛНУ, 2021. 198 с.

33. Нестеренко О. В. Основи технічної кібербезпеки. Харків : ХНУРЕ, 2020. 156 с.

34. Степаненко І. В., Мороз Ю. В. Комп'ютерне моделювання в системах безпеки. Київ : ВЦ «Академія», 2020. 174 с.

35. ISO/IEC 19794-5:2011. Information technology – Biometric data interchange formats – Part 5: Face image data. Geneva : ISO, 2011. 44 p.

36. ISO/IEC 2382:2015. Information technology – Vocabulary. Geneva : ISO, 2015. 456 p.

37. NIST Special Publication 500-290. Face Recognition Vendor Test (FRVT). – Gaithersburg : NIST, 2020. – 63 p.

38. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання. Вид. офіц. Київ : Мінекономрозвитку України, 2015. 16 с. (Інформація та документація).

39. ДСТУ 7152:2021. Видання. Оформлення публікацій у журналах і збірниках. Вид. офіц. Київ : Мінекономіки України, 2021. 16 с.

40. ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вид. офіц. Київ : УкрНДНЦ, 2016. 44 с.
41. ISO/IEC 30107-3:2017. Biometric presentation attack detection – Part 3: Testing and reporting. Geneva : ISO, 2017. 28 p.
42. ISO/IEC 19795-1:2006. Information technology – Biometric performance testing and reporting – Part 1: Principles and framework. Geneva : ISO, 2006. 60 p.
43. ДСТУ 3634:2021. Системи технічного захисту інформації. Терміни та визначення. Вид. офіц. Київ : УкрНДНЦ, 2021. 24 с.
44. Литвиненко І. В. Безпека інформаційних технологій : навч. посіб. Львів : ЛНУ, 2020. 208 с.
45. Бабенко А. І., Лисенко В. А. Програмне забезпечення систем штучного інтелекту. Київ : КНЕУ, 2020. 192 с.
46. Буряк Н. С. Методи виявлення аномалій у кіберпросторі : монографія. Харків : ХНУРЕ, 2021. 240 с.
47. Карпенко І. М. Інформаційні системи і технології у безпеці : навч. посіб. Київ : НАВС, 2020. 176 с.
48. Петренко Ю. І. Програмування нейронних мереж : підручник. Львів : ЛНУ, 2021. 264 с.
49. Дударенко В. В., Гринюк Б. О. Розпізнавання образів у прикладних задачах : навч. посіб. Київ : Техніка, 2020. 158 с.
50. Калюжний Р. І. Нейронні мережі в задачах класифікації. Тернопіль : ТНТУ, 2021. 212 с.
51. Місюра М. Д., Шкарупило В. В. Інформаційна безпека та ідентифікація : навч. посіб. Харків : УПА, 2021. 192 с.
52. Захарченко В. О. Біометрія в інформаційних системах. Одеса : ОНПУ, 2020. 205 с.
53. Дахно І. І. Кібербезпека у відкритих системах : монографія. Київ : Ліра-К, 2021. 220 с.
54. Кривенко А. Ю., Мельник Т. П. Технології штучного інтелекту : навч.

посіб. Київ : КНЕУ, 2021. 180 с.

55. Смолій В. В., Савицька Я. А., Місюра М. Д. Розпізнавання емоційних станів за зображенням обличчя. Харків : ХНУРЕ, 2021. 164 с.

56. Веретюк О. М. Ідентифікація особи за біометричними характеристиками : монографія. Київ : Університет, 2020. 248 с.

57. Грабар І. В. Математичне забезпечення систем розпізнавання. Тернопіль : ТНТУ, 2020. 188 с.

58. Юрченко А. В. Теорія нейронних мереж : навч. посіб. Київ : КНЕУ, 2020. 196 с.

59. Куценко С. І., Федоренко А. Ю. Обробка зображень та відеопотоків : навч. посіб. Харків : ХНУРЕ, 2021. 210 с.

60. Якимчук І. В. Автоматизовані системи контролю доступу : підручник. Львів : ЛНУ, 2021. 172 с.

61. Піддубна Н. І. Алгоритми обробки зображень в системах візуального контролю. Київ : Техніка, 2021. 186 с.

62. Павленко О. М. Методи сегментації зображень у медичних системах. Харків : ХНУРЕ, 2020. 174 с.

63. Сергієнко В. А. Паралельні обчислення та їх використання. Київ : Наука, 2020. 210 с.

64. Шевченко О. А. Методи глибокого навчання у розпізнаванні образів. Дніпро : ДНУ, 2021. 192 с.

65. Герасименко А. П., Литвин І. Ю. Статистичне розпізнавання образів : навч. посіб. Харків : УПА, 2021. 184 с.

66. Грицай Н. О. Комп'ютерний зір : концепції та алгоритми. Тернопіль : ТНТУ, 2021. 160 с.

67. Коваленко Д. А. Оптимізація алгоритмів класифікації зображень. Київ : КНЕУ, 2020. 170 с.

68. Федорук А. М. Застосування штучного інтелекту в охоронних системах. Одеса : ОНАХТ, 2021. 200 с.

69. Мірошниченко В. В. Біометричні методи автентифікації. Львів : ЛНУ,

2020. 218 с.

70. Дяченко С. В. Основи побудови нейромережових моделей. Київ : Видавництво Ліра-К, 2021. 230 с.

71. Калінін О. В. Моделі глибокого навчання для аналізу відео. Харків : ХНУРЕ, 2020. 216 с.

72. Кузьменко С. О. Апаратне прискорення нейронних мереж. Київ : КНЕУ, 2021. 194 с.

73. Головка О. М. Нейронні мережі у біомедичних задачах. Дніпро : ДНУ, 2020. 200 с.

74. Іванова Л. М. Методи виявлення облич у реальному часі. Одеса : ОНПУ, 2021. 180 с.

75. Кравченко В. Ю. Ідентифікація об'єктів на відеопотоці. Харків : ХНУРЕ, 2020. 196 с.

76. Євтушенко Т. С. Застосування Python у біометричних системах. Київ : КНЕУ, 2020. 202 с.

77. Шарій В. М. Теоретичні основи цифрової обробки сигналів. Львів : ЛНУ, 2021. 208 с.

78. Максименко О. В. Автоматизовані системи обробки зображень. Тернопіль : ТНТУ, 2020. 178 с.

79. Снігур В. І. Штучний інтелект у технічній охороні. Київ : Наука, 2021. 204 с.

80. Петренко Т. С. Розпізнавання облич у мобільних пристроях. Харків : ХНУРЕ, 2021. 160 с.

81. Васильченко Ю. Б. Методи покращення якості зображень. Київ : Видавництво Ліра-К, 2020. 182 с.

82. Литвиненко Н. А. Безпека біометричних даних. Одеса : ОНАХТ, 2021. 210 с.

83. Назаренко Д. П. Системи виявлення вторгнень з використанням нейронних мереж. Львів : ЛНУ, 2021. 218 с.

## ДОДАТОК А

(обов'язковий)

### КОПІЯ ТЕЗ ДОПОВІДІ НА МІЖНАРОДНІЙ КОНФЕРЕНЦІЇ

*Секція «Інформаційно-аналітичні та інформаційно-керувачі системи»*

УДК 004.8

**Б.О. Мартинюк, В.М. Григ**  
Хмельницької національний університет  
gr.volodymyr2018@gmail.com

#### СПЕЦІАЛІЗОВАНА КОМП'ЮТЕРНА СИСТЕМА ІДЕНТИФІКАЦІЇ ОСОБИ НА БАЗІ НЕЙРОННОЇ МЕРЕЖІ

В роботі розроблено спеціалізовану комп'ютерну систему ідентифікації особи на базі нейронної мережі. Проведено аналіз предметної області подібних систем та обґрунтовано їх актуальність у сфері розпізнавання обличч, об'єктів та тексту. У процесі розроблення вибрано мову програмування Python із використанням фреймворку Flask для реалізації серверної частини, а також технології HTML, CSS. Функціональність системи охоплює розпізнавання обличчя за допомогою алгоритмів комп'ютерного зору, аналіз об'єктів на зображеннях та ідентифікацію тексту за допомогою машинного навчання. Тестування та налагодження програмного забезпечення проводилося на локальному сервері.

У сучасному світі технології ідентифікації особи відіграють важливу роль у забезпеченні безпеки, контролю доступу та автоматизації різних процесів. Традиційні методи ідентифікації, такі як введення паролів або використання документів, часто є недостатньо надійними, оскільки їх можна підробити або викрасти. У зв'язку з цим все більшої популярності набувають системи біометричної ідентифікації, які базуються на аналізі унікальних фізичних характеристик людини, зокрема обличчя.

Застосування нейронних мереж та алгоритмів глибокого навчання у сфері розпізнавання обличчя значно підвищує точність і швидкість ідентифікації [1]. Однак, багато існуючих рішень мають певні обмеження, такі як залежність від умов освітлення, кута нахилу обличчя або низька продуктивність при обробці великої кількості зображень. Тому розробка ефектвної, гнучкої та доступної системи ідентифікації особи є актуальним завданням.

Розроблена система дозволяє не лише розпізнавати обличчя, а й аналізувати об'єкти та текст на зображенні, що значно розширює її можливості у сфері автоматизації та безпеки. Вибір Python та Flask як основної технологічної платформи дозволяє реалізувати ефективну серверну частину, а використання OpenCV та методів машинного навчання підвищує точність розпізнавання.

Метою роботи є розробка спеціалізованої комп'ютерної системи ідентифікації особи на базі нейронної мережі, яка забезпечуватиме автоматичне розпізнавання обличчя, об'єктів та тексту на зображеннях із високою точністю та швидкістю обробки. Для досягнення цієї мети необхідно виконати такі завдання: проаналізувати сучасні методи ідентифікації особи та визначити їхні переваги та недоліки; обґрунтувати вибір середовища розробки, мов програмування та бібліотек; реалізувати серверну частину веб-додатка на базі Flask; інтегрувати методи машинного навчання для розпізнавання обличчя, об'єктів та тексту; створити зручний веб-інтерфейс з можливістю завантаження зображень та перегляду результатів; протестувати систему на різних типах зображень, забезпечивши коректність розпізнавання та мінімізацію помилок; налаштувати систему для подальшого використання та можливості інтеграції в інші сервіси.

На початковому етапі роботи проведено аналіз сучасних методів ідентифікації особи та розпізнавання об'єктів. Досліджено переваги та недоліки існуючих рішень, зокрема біометричних систем на основі відбитків пальців, райдужної оболонки ока та розпізнавання обличчя. Було визначено, що системи на основі розпізнавання обличчя є найбільш зручними, оскільки вони не потребують фізичного контакту з пристроєм і можуть працювати в реальному часі [2].

З огляду на це, було прийнято рішення використовувати нейронні мережі для аналізу зображень. Для реалізації серверної частини обрано мову програмування Python та веб-фреймворк Flask, що забезпечує легкість інтеграції алгоритмів машинного навчання в веб-

додаток. Для роботи із зображеннями використано OpenCV, а для створення та тренування нейронної мережі застосовано PyTorch.

Серверна частина реалізована на Flask, який обробляє запити користувачів, приймає зображення та передає їх у нейромережу для аналізу. Алгоритм роботи сервера:

1. Користувач завантажує зображення через веб-інтерфейс.
2. Сервер отримує файл, обробляє його та передає до модуля розпізнавання.
3. Нейронна мережа аналізує зображення, порівнює його з відомими шаблонами та повертає результат.
4. Вебдодаток відображає користувачеві відповідь із результатом розпізнавання.

Для зберігання та обробки зображень використано Werkzeug для безпечного завантаження файлів, а також механізм попередньої обробки, що включає зміну розміру, нормалізацію та конвертацію в формат, придатний для аналізу нейромережею.

Для розпізнавання обличчя використано попередньо навчену модель на базі ResNet, адаптовану під задачу класифікації відомих та невідомих обличчя. Модель отримує зображення, перетворює його в набір числових характеристик (вектор ознак) і порівнює з базою відомих осіб. Якщо схожість перевищує певний поріг, особи вважається відомою, в іншому випадку система фіксує невідоме обличчя [3].

Для розпізнавання об'єктів у зображеннях застосовано модель YOLO (You Only Look Once), що дозволяє знаходити та класифікувати об'єкти в реальному часі. Це розширює функціонал системи, дозволяючи аналізувати не лише обличчя, а й будь-які інші елементи на зображенні.

Для розпізнавання тексту використано технологію OCR (Optical Character Recognition), яка дозволяє ідентифікувати символи на зображеннях і конвертувати їх у текст. Це корисно для аналізу документів або витягування інформації з фотографій.

Для вебінтерфейсу створено зручний дизайн, що включає:

- Форму завантаження зображень, де користувач може обрати тип розпізнавання: обличчя, об'єкти або текст.
- Випадаючий список вибору категорії, що дозволяє налаштувати аналіз під конкретний тип зображення.
- Кнопку завантаження, яка ініціює процес обробки та відображає результат у полі "Результат розпізнавання".
- Гарний стиль з градієнтним фоном та адаптивним дизайном, що покращує візуальне сприйняття та забезпечує зручність використання на різних пристроях.

На рис. 1 продемонстровано вигляд інтерфейсу:

Рисунок 1 – Інтерфейс програми для розпізнавання обличчя

Під час тестування перевірено точність розпізнавання на різних типах зображень. Було проведено серію експериментів із фотографіями в різних умовах освітлення, із зміною ракурсу та додаванням шумів. Оптимізовано роботу нейромережі шляхом корекції гіперпараметрів та використавши попередньо оброблених даних для навчання моделі (рис.2):

Також перевірено коректність обробки зображень у різних форматах (JPEG, PNG, BMP) та протестовано стабільність роботи сервера при одночасному завантаженні великої кількості запитів [4].

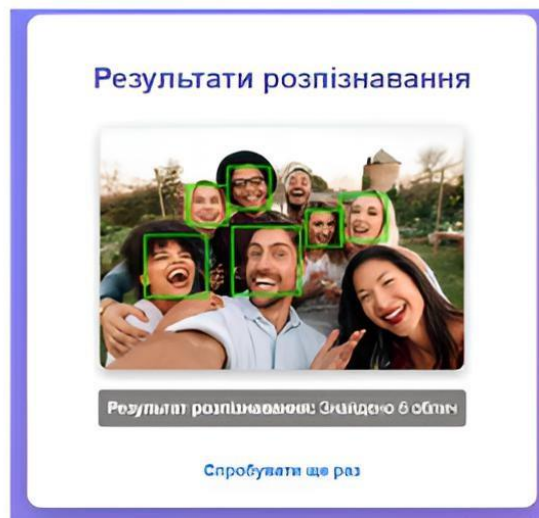


Рисунок 2 – Результат розпізнавання кількості людей на фото

В результаті виконаної роботи створено ефективну систему ідентифікації особи на основі нейронної мережі, яка дозволяє розпізнавати обличчя, об'єкти та текст. Реалізовано вебінтерфейс, що забезпечує зручний доступ до функціоналу, а також протестовано систему на реальних даних для оцінки точності розпізнавання. Розроблена система має можливість подальшої інтеграції в системи контролю доступу, безпеки або автоматизованої обробки зображень.

#### ЛІТЕРАТУРА:

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: MIT Press, 2016. 800 p.
2. Rosebrock A. Deep Learning for Computer Vision with Python. PyImageSearch, 2019. 640 p.
3. Guo G., Zhang N. A Survey on Deep Learning-Based Face Recognition. Computer Vision and Image Understanding, Vol. 189, 2019, Article ID 102805. DOI: 10.1016/j.cviu.2019.102805.
4. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. arXiv preprint, 2018. arXiv:1804.02767.

## ДОДАТОК Б

# ПРЕЗЕНТАЦІЯ ДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

Кваліфікаційна робота на здобуття освітнього ступеня магістра

## *Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі*

Підготував:  
студент групи КІ2м-23-3, Богдан МАРТИНЮК  
Науковий керівник:  
Кандидат т.н., доцент  
комп'ютерної інженерії та  
електроніки, Володимир ГРИГА

Хмельницький, 2025 рік



## Актуальність, мета, об'єкт, предмет

2

- Актуальність цієї роботи обумовлена зростаючою потребою в удосконаленні систем ідентифікації особи, що відповідають високим вимогам безпеки та ефективності. З традиційними методами, такими як паролі чи картки доступу, все частіше виникають проблеми з безпекою, оскільки вони можуть бути вкрадені, забуті або підроблені. Нейронні мережі, у свою чергу, дозволяють значно підвищити точність ідентифікації завдяки здатності до навчання на великих наборах даних та їх здатності адаптуватися до змінних умов.
- Об'єктом дослідження є процеси ідентифікації особи на основі нейронних мереж у системах безпеки, зокрема в спеціалізованих комп'ютерних системах для забезпечення доступу до захищених ресурсів.
- Предметом дослідження є система ідентифікації особи на основі нейронної мережі для забезпечення безпеки в автоматизованих системах доступу.
- Метою кваліфікаційної роботи магістра є підвищення ефективності та точності ідентифікації особи в системах безпеки за допомогою використання нейронних мереж для розпізнавання особливих біометричних характеристик, таких як обличчя, відбитки пальців або райдужка ока.
- Для розв'язання поставлених задач використовувалися математичні та аналітичні методи дослідження, комп'ютерне моделювання, методи оцінки ефективності, а також методи оцінки точності та швидкості системи ідентифікації.

## Наукова новизна

3

- Наукова новизна отриманих результатів:
- Набула подальшого розвитку базова модель системи ідентифікації особи на основі нейронної мережі, яка включає використання сучасних архітектур глибоких нейронних мереж для покращення точності та швидкості ідентифікації. Це дозволило розробити нові методи оптимізації навчання нейронної мережі для зменшення помилок ідентифікації в реальних умовах.
- Набув подальшого розвитку метод інтеграції нейронних мереж в мультиагентні системи для підвищення ефективності ідентифікації особи, що дозволяє значно зменшити час реагування на запити та підвищити рівень безпеки.
- Практична цінність роботи полягає в тому, що запропонована система ідентифікації особи на базі нейронної мережі може бути інтегрована в існуючі системи контролю доступу в різних галузях, таких як фінансові установи, державні органи або в систему Інтернету речей. Це дозволить підвищити ефективність доступу до захищених ресурсів та зменшити ризик несанкціонованого доступу до важливих даних.

## Аналіз існуючих рішень

3

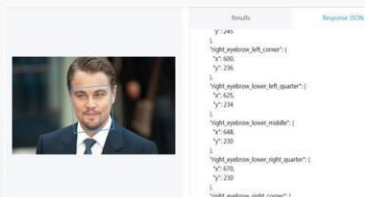
Класифікація методів розпізнавання (геометричні, фотометричні)

API: Azure Face, Google Cloud Vision, FaceID

Таблиця порівняння (точність, швидкодія, функціонал)



Виявленні обличчя на фото



Визначення координат та атрибутів



Порівняння людських образів



Порівняння подібності людських образів



Оцінка достовірності для кожної емоції



Пошук подібних людських образів

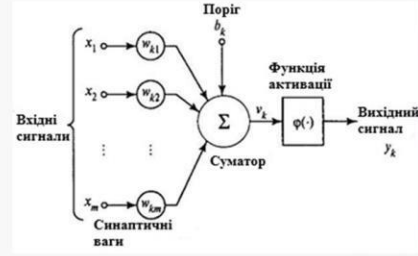
# Вибір методу та нейронна мережа

Обґрунтування вибору CNN

Активні функції: sigmoid, tanh, ReLU



Загальний вигляд структури перцептрону



Формальна модель нейрону

$$f(w_1x_1 + w_2x_2 + \dots + w_mx_m + b)$$

Формула виходу нейрона



# Архітектура системи ідентифікації

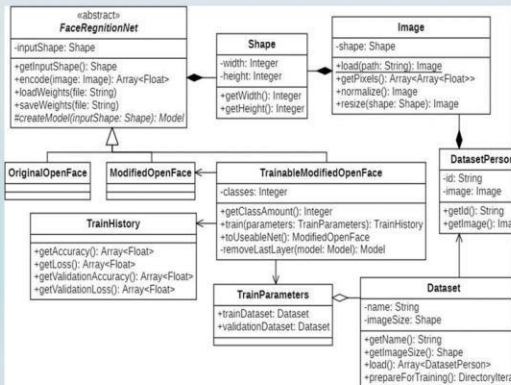
Етапи: детекція → попередня обробка → розпізнавання → вивід результату

Мультиагентний підхід

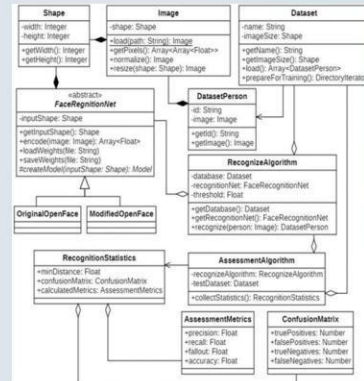


Архітектура розпізнавання людських образів

# Діаграми класів для машинного навчання

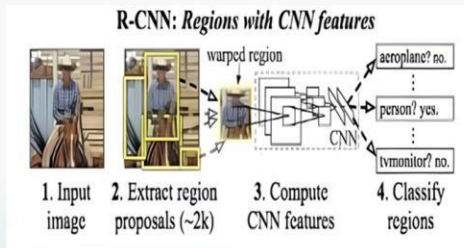


Діаграма загальних класів та класів для реалізації навчання модифікованої мережі розпізнавання людських образів

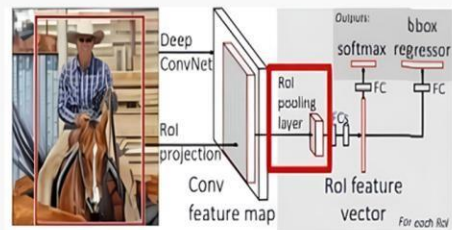


Діаграма класів для реалізації алгоритмів розпізнавання людських образів та оцінювання нейронної мережі

# Розробка моделі



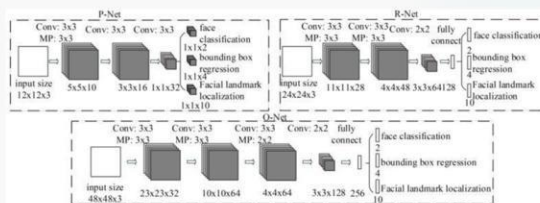
Алгоритм роботи R-CNN



Архітектура мережі Fast R-CNN

$$\|f(x^a) - f(x^p)\|^2 + a < \|f(x^a) - f(x^n)\|^2,$$

Формула функції втрат

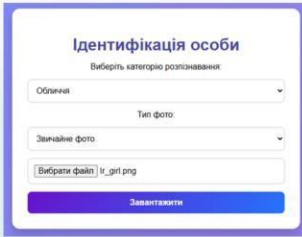


Архітектура мережі MTCNN

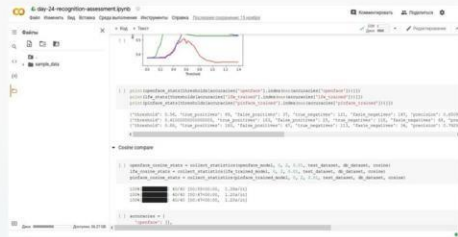
# Реалізація програмного середовища

8

Використано: Python, Flask, OpenCV, TensorFlow  
Інтерфейс системи — веб-додаток  
Середовище: Google Colab, VS Code



Інтерфейс програми для розпізнавання обличчя



Вигляд інтерфейсу Google Colab



NVIDIA CUDA

```
# Перехід у директорію проекту
cd D:/Projects/face-recognition-cnn

# Створення та активація віртуального оточення у поточній директорії
python -m venv ;
scripts/activate

# Встановлення залежностей через Pip
pip install tensorflow==2.4.1
pip install keras==2.4.3
pip install numpy==1.19.5
pip install matplotlib==3.4.0
pip install jupyterlab==3.0.12
pip install opencv-python==4.5.1.48
```

Команди процесу налаштування середовища розробки

# Експериментальна оцінка

9

Матриця заплутаності

	Позитивні	Негативні
Позитивні	Істинно позитивні (True Positive, TP)	Помилково позитивні (False Positive, FP)
Негативні	Помилково негативні (False Negative, FN)	Істинно негативні (True Negative, TN)

**Початок:**  
 Дати шлях до директорії датасету;  
 Дати вірогідність поповнення в базу даних;  
 список піддиректорій = завантажити із шляху до директорії датасету тестовий датасет = пустий асоціативний масив;  
 датасет БД = пустий асоціативний масив;  
 Для кожного імені людини в списку піддиректорій:  
 тестовий датасет[им'я людини] = пустий масив;  
 випадкове число = згенерувати випадкове число в межах [0; 1);  
 Якщо випадкове число < вірогідності поповнення в базу даних:  
 додати в базу даних = True;  
 датасет БД[им'я людини] = пустий масив;  
 Інакше:  
 додати в базу даних = False;  
 Для кожного зображення в піддиректорії імені людини:  
 Якщо додати в базу даних:  
 додати зображення в масив датасет БД[им'я людини];  
 додати в базу даних = False;  
 Інакше:  
 додати зображення в масив тестовий датасет[им'я людини];

**Кінець**

Алгоритм генерації датасетів для оцінки мереж

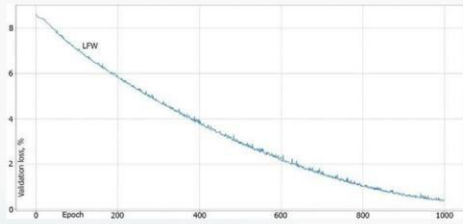


Приклади зображень із датасету PinsFace

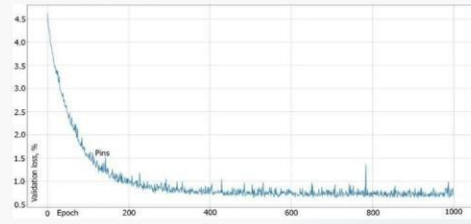


Приклади зображень із датасету ORL Faces.

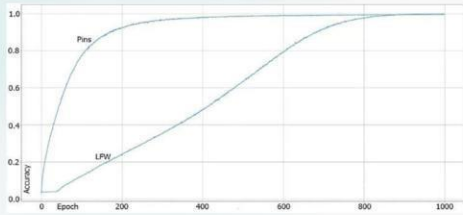
# Експериментальна оцінка



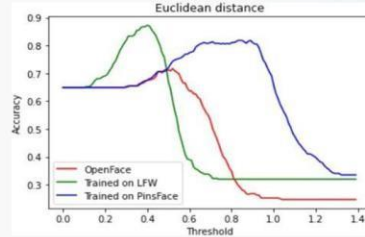
Валідація на датасеті LFW



Валідація на датасеті PinsFace

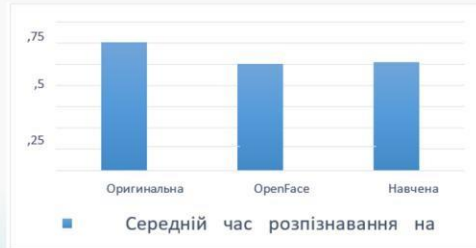


Точність навчання на використаних датасетах



Залежність акуратності розпізнавання від порога порівняння векторів по Евклідовій відстані

# Результати та аналіз робробки



Середній час розпізнавання на одну ітерацію

```
Epoch 1/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 2/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 3/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 4/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 5/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 6/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 7/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 8/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 9/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 10/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 11/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 12/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 13/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 14/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 15/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 16/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 17/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 18/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 19/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
Epoch 20/2000 [.....] - loss: 6.9206/step - loss: 6.4287 - accuracy: 0.8181 - val_loss: 0.5949 - val_accuracy: 0.8025
```

Процес навчання мереж на Tensorflow з Keras

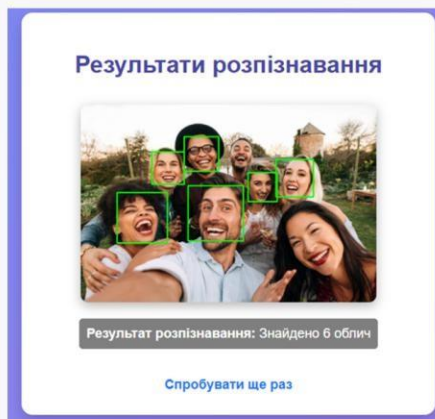
Тип машини	Робоча станція	Виділена машина Google Colab
CPU	6,4 годин	13 годин
GPU	1,3 годин	2,3 годин
TPU		13,3 годин

Приблизний час навчання на робочій станції ПМІ та виділеній машині Google Colab на CPU, GPU та TPU



## Результати та аналіз робробки

12



Фінальний результат розпізнавання кількості людей на фото



## Висновки

13

У процесі виконання магістерської кваліфікаційної роботи першочергово було здійснено глибокий аналіз предметної області, пов'язаної з автоматичним виявленням та розпізнаванням людських образів.

1. У першому розділі проведено аналіз сучасного стану технологій ідентифікації особи, особливу увагу приділено нейронним мережам як основі для розпізнавання обличчя. Було розглянуто переваги біометричних методів порівняно з традиційними, а також проаналізовано найефективніші програмні засоби, такі як Azure Face API, Face ID і Cloud Vision API. Встановлено, що найбільш перспективним напрямком є застосування згорткових нейронних мереж для ідентифікації осіб у реальному часі.
2. У другому розділі обґрунтовано вибір нейронних мереж як основного інструменту для побудови системи розпізнавання обличчя. Розглянуто архітектуру системи, визначено структуру та особливості згорткових і повноз'язних мереж, активаційні функції, алгоритми навчання та оптимізації. Запропоновано концептуальну модель системи, що враховує основні параметри зображень, які впливають на точність розпізнавання (освітлення, кут огляду тощо).
3. У третьому розділі розроблено архітектуру та математичну модель системи ідентифікації особи. Визначено критерії оптимізації – точність, швидкість і надійність. Запропоновано використання мультиагентної системи для інтеграції з нейронними мережами, що дозволяє підвищити ефективність функціонування системи в реальних умовах.
4. У четвертому розділі реалізовано програмну частину розробленої системи, підготовлено середовище для навчання нейромережі, розроблено класи для тренування та тестування, а також реалізовано алгоритми оцінювання продуктивності. Проведено експериментальні дослідження, які підтвердили, що запропонована система забезпечує вищу точність та стійкість до змін зовнішніх умов порівняно з іншими методами. Результати експериментів засвідчили переваги обраного підходу.

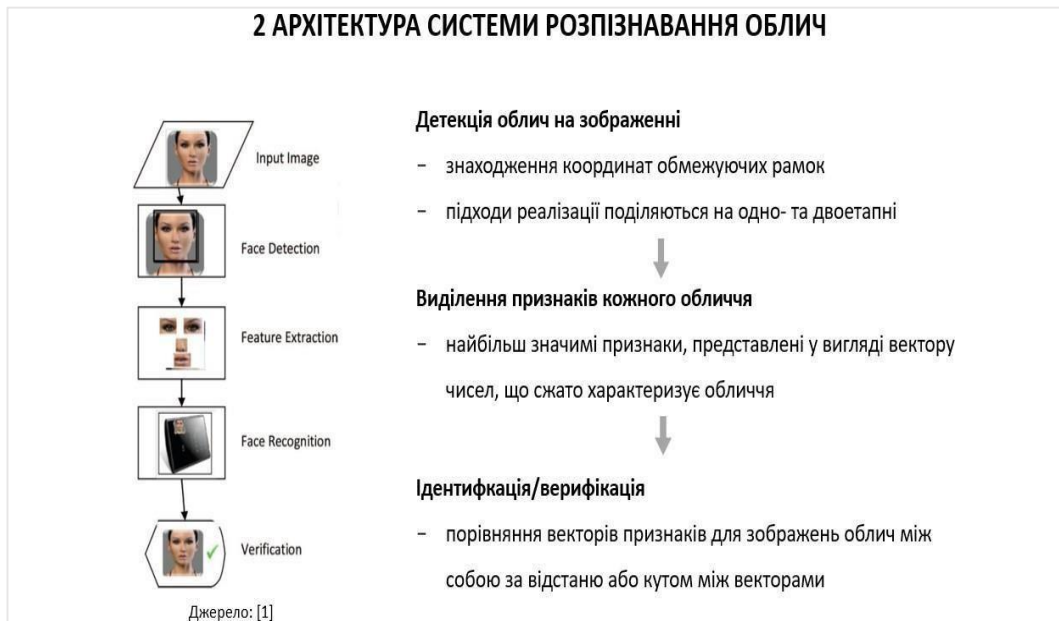


*Дякую за увагу!*



## ДОДАТОК В

### АРХІТЕКТУРА РОЗПІЗНАВАННЯ ЛЮДСЬКИХ ОБРАЗІВ



## ДОДАТОК Г

### ВИДІЛЕННЯ ПРИЗНАКІВ

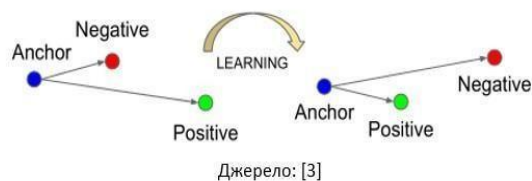
#### 3 ВИДІЛЕННЯ ПРИЗНАКІВ

**Головна задача:** навчити нейронну мережу для зображень однієї людини видавати схожі вектори признаков, а для різних людей – різні вектори.

##### Існуючі підходи

1. Використання функції помилки Triplet Loss  
(використовується у мережі FaceNet и OpenFace)

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0\right)$$



2. Навчання класифікатору

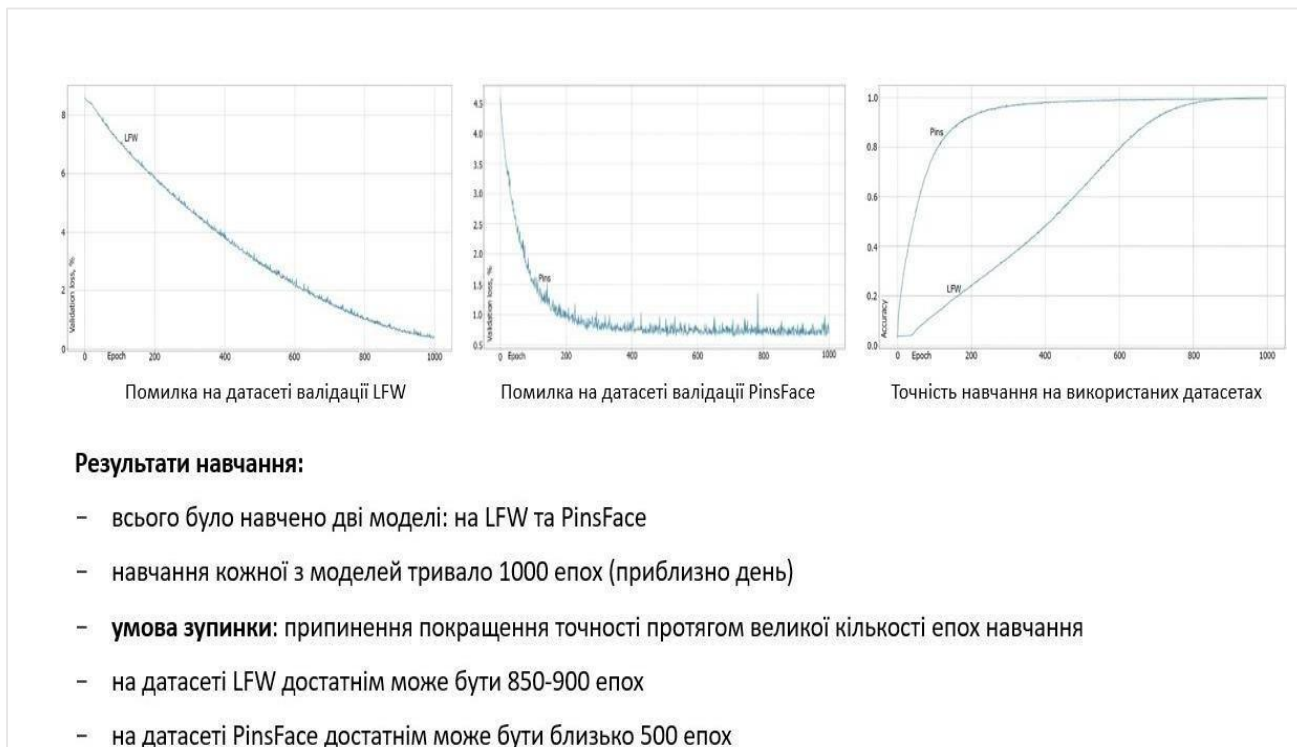
- навчити мережу класифікувати фіксований набір людей
- після навчання видалити останній (класифікуючий) шар
- мережа, що залишилась, вже вміє правильно виділяти признаки

## ДОДАТОК Д МОДИФІКАЦІЯ МЕРЕЖІ



## ДОДАТОК Е

### РОЗРОБКА ТА НАВЧАННЯ МОДЕЛЕЙ



## ДОДАТОК Ж

### ТЕСТУВАННЯ МОДЕЛЕЙ

#### МЕТРИКИ ОЦІНКИ – CONFUSION MATRIX

	Істинно позитивні	Істинно негативні
Позитивне розпізнавання	True Positive (TP) Розпізнавання вірно	False Positive (FP) Розпізнавання невірно
Негативне розпізнавання	False Negative (FN) Розпізнавання невірно	True Negative (TN) Розпізнавання вірно

- ✓ True Positive приклади мережа вірно співвіднесла до людей в БД
- ✓ True Negative приклади мережа не співвіднесла до людей, яких немає в БД
- ✗ False Positive приклади мережа невірно співвіднесла до людей в БД
- ✗ False Negative приклади мережа не співвіднесла до людей, що є в БД

$$\text{Загальна акуратність розпізнавання} = \frac{TP+TN}{TP+TN+FP+FN}$$

Джерело: [1]

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Богдан МАРТИНЮК

**Співавтор:**

**Назва:** Мартинюк\_Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 0.8%

**Коефіцієнт подібності 2:** 0.1%

**Мікропробіли:** 0

**Заміна букв:** 2

**Інтервали:** 0

**Білі знаки:** 1

**Дата створення звіту:** 2025-05-16 15:32:28.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-05-16

Дата

Доцент Андрій Нічепорук

експерт

## Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 7%

ID: 241284 Title: МКР Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі Added in a DB: 2025-05-16 Authors: Богдан МАРТИНЮК Heads: Володимир ГРИГА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	133218	1047	1157 (1%)	20 (2%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Богдан МАРТИНЮК

Тема: Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 98

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано систему профілювання вразливостей при керуванні розумним будинком

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі виконано огляд існуючих методів і технологій ідентифікації особи, зокрема через нейронні мережі. Проаналізовано переваги та недоліки традиційних методів ідентифікації (паролі, картки) порівняно з методами біометричної ідентифікації. У другому розділі розроблено абстрактну модель для системи ідентифікації особи. Ця модель включає використання різних типів нейронних мереж (конволюційних, рекурентних) для обробки зображень та інших біометричних даних. У третьому розділі були визначені основні критерії для оптимізації роботи системи ідентифікації особи. Для цього була розроблена математична модель, що включає вагові коефіцієнти для різних факторів, таких як точність, швидкість і надійність. Запропоновано архітектуру системи, яка включає використання нейронних мереж для розпізнавання образів і інтеграцію цих мереж в мультиагентну систему для забезпечення високої ефективності ідентифікації. У четвертому розділі була проведена оцінка ефективності розробленої системи ідентифікації. Проведено порівняння результатів роботи системи, заснованої на базовій моделі і нейронній мережі, з іншими відомими методами

ідентифікації. Було виявлено, що запропонована система забезпечує більш високу точність і швидкість ідентифікації, а також менше піддається впливу факторів навколишнього середовища, таких як зміни освітлення чи кути огляду.

4. Позитивні сторони роботи: Запропонована система ідентифікації особи на базі нейронної мережі може бути інтегрована в існуючі системи контролю доступу в різних галузях, таких як фінансові установи, державні органи або в систему Інтернету речей. Це дозволить підвищити ефективність доступу до захищених ресурсів та зменшити ризик несанкціонованого доступу до важливих даних.

5. Негативні сторони роботи: В роботі наявні аналітичні вирази, які не мають повного розшифрування коефіцієнтів. В тексті роботи зустрічаються англійські скорочення, які не мають пояснення у переліку скорочень

6. Оцінка графічного оформлення та пояснювальної записки роботи: \_\_\_\_\_  
Пояснювальна записка та презентаційний матеріал оформлені згідно вимог та поставленого завдання.

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: Відсутні

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «задовільно» (D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_  
д.т.н., професор, Мартинюк В.В., завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

“ 20 травня ” \_\_\_\_\_ 2025р.

Завідувачу кафедри КПС  
доктору філософії, доценту  
Ользі ПАВЛОВІЙ

Мартинюка Богдана Олександровича

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-23-3

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

20 травня 2025 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі

Автор: Мартинюк Богдан Олександрович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Грига Володимир Михайлович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

запозичення розміщені в розділах є збіг з звітом з науково-дослідної практики автора Богдана Мартинюка "Спеціалізована комп'ютерна система ідентифікації особи на базі нейронної мережі", який було додано в репозитарій ХНУ 21 березня 2025 року;

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 0.8% і адресується до 16 першоджерел; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС

Володимир ГРИГА

Олег САВЕНКО

Ольга ПАВЛОВА