

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

дискретного аналізу та інтелектуальних систем

(повна назва кафедри)

## Дипломна робота

РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ІЗ ВИКОРИСТАННЯМ ШТУЧНОЇ  
НЕЙРОННОЇ МЕРЕЖІ

Виконала: студентка групи ПМі-43с  
спеціальності

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Матуш Вікторія Валеріївна

(підпис)

(прізвище та ініціали)

Керівник

Колос Надія Мирославівна

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

2023

***ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА***

**Факультет Прикладної математики та інформатики**

**Кафедра Дискретного аналізу та інтелектуальних систем**

**Спеціальність 122 «Комп'ютерні науки»**

(шифр і назва)

**«ЗАТВЕРДЖУЮ»**

**Завідувач кафедри Притула М.М.**

**"31" серпня 2022 року**

**ЗАВДАННЯ**

**НА ДИПЛОМНУ У РОБОТУ СТУДЕНТУ**

**Матуш Вікторії Валеріївній**

(прізвище, ім'я, по батькові)

1. Тема роботи **«Розпізнавання обличчя з використанням штучної нейронної мережі»**,

керівник роботи **Колос Надія Мирославівна, кандидат фіз.-мат. наук**

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

затверджені Вченою радою факультету від **"13" вересня 2022 року № 15**

2. Строк подання студентом роботи **13.06.2023р.**

3. Вихідні дані до роботи **мова програмування Python, інтегроване середовище розробки (IDE) PyCharm та бібліотеки, такі як os.path, datetime, subprocess, tkinter, cv2 (OpenCV), util і PIL (Python Imaging Library), інтернет ресурси**

4. Зміст дипломної роботи (перелік питань, які потрібно розробити) **розробка системи обліку відвідуваності з використанням технології розпізнавання обличчя, що дозволяє автоматизувати та точно відстежувати присутність. Крім того, створення зручного графічного інтерфейсу користувача (GUI), який сприяє простій навігації та взаємодії з системою, покращує загальний досвід користувача та зручність використання.**

## 5. Консультанти розділів роботи

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

6. Дата видачі завдання **31 серпня 2022 р.**

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз та постановка задачі	18.10.2022	
2.	Вивчення літератури по обраній темі	03.11.2022	
3.	Огляд існуючих програмних рішень задачі	20.12.2022	
4.	Проектування архітектури та структури системи	05.01.2023	
5.	Реалізація основних функцій системи	10.01.2023	
6.	Створення зручного графічного інтерфейсу користувача	17.01.2023	
7.	Реалізація додаткових функцій та покращень	04.03.2023	
8.	Тестування та забезпечення якості	28.04.2023	
9.	Оформлення дипломної роботи	31.05.2023	

Студент \_\_\_\_\_ Матуш В.В.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Колос Н.М.  
( підпис ) (прізвище та ініціали)

## РЕФЕРАТ

У даній бакалаврській дипломній роботі було досліджено тему «Розпізнавання обличчя з використанням штучної нейронної мережі».

Головною метою даної роботи є створення системи розпізнавання обличчя за допомогою штучної нейронної мережі з точною відповідністю та ідентифікацією осіб на різних відеокадрах. Шляхом розробки автоматизованої системи, що здатна точно підтверджувати ідентичність осіб на основі їх обличчя. Метою є полегшити та зробити більш ефективним процес аутентифікації.

У ході роботи було досліджено програмні рішення та технології, які застосовуються для ідентифікації обличчя. Основний аспект даної роботи полягає у розробці системи розпізнавання обличчя з використанням штучної нейронної мережі.

Результатом даної роботи є система, що пропонує автоматизоване відстеження присутності за допомогою технології розпізнавання облич. Система включає такі функції, як реєстрація користувачів, функціонал входу в систему та можливість зберігати записи про присутність у окремих текстових файлах. Завдяки своєму зручному інтерфейсу та використанню різноманітних бібліотек та інструментів, система надає практичне рішення для спрощення процесу аутентифікації та управління присутністю.

Для розробки застосунку було використано мову програмування Python.

Обсяг дипломної роботи 40 сторінок, 10 використаних джерел.

Ключові слова: глибинне навчання, розпізнавання обличчя, згортова нейронна мережа.

<b>ВСТУП.....</b>	<b>6</b>
Постановка проблеми та цілі дослідження.....	8
<b>Розділ 1: Що таке нейронні мережі - історія появи, типи, відмінності... 9</b>	<b>9</b>
1.1 Біологічні нейрони.....	9
1.2 Штучні нейронні мережі.....	10
1.3 Будова нейронної мережі.....	12
1.4 Функції активації.....	13
1.5 Навчання нейронних мереж.....	15
1.6 Типи нейронних мереж.....	17
1.7 Використана модель мережі.....	22
<b>Розділ 2: Методологія.....</b>	<b>24</b>
2.1 Виявлення обличчя, розпізнавання обличчя, верифікація обличчя..	24
2.2 Огляд методів для виявлення обличчя, вилучення ознак та класифікації.....	25
<b>Розділ 3: Реалізація.....</b>	<b>28</b>
3.1 Розробка UI/UX дизайну в середовищі Figma.....	28
3.2 Використані технології.....	30
3.3 Опис роботи.....	31
<b>Висновки.....</b>	<b>37</b>
<b>Список використаної літератури.....</b>	<b>39</b>

## ВСТУП

Розпізнавання обличчя - це швидкозростаюча галузь комп'ютерного зору та штучного інтелекту, яка має на меті автоматичне ідентифікування та аутентифікацію осіб на основі рис їх облич. Вона здобула значну увагу та важливість завдяки широкому спектру застосувань у різних галузях, включаючи системи безпеки, нагляд, контроль доступу, біометрію та взаємодію людина-комп'ютер.

Здатність розпізнавати та відрізнити обличчя є основною навичкою для людей, і відтворення цієї здатності в машинах завжди було складним завданням. Традиційні техніки розпізнавання обличчя ґрунтувалися на створенні вручну розроблених ознак та статистичних моделях, які часто мали проблеми з варіаціями пози чи освітлення. Однак, завдяки останнім досягненням у глибинному навчанні та нейронних мережах, розпізнавання обличчя відзначилося значним прогресом та досягло небаченої точності.

Нейронні мережі зробили можливим втілення технології розпізнавання обличчя, дозволяючи автоматичне вивчення відокремлювальних ознак безпосередньо з сирого зображення обличчя. Ці мережі використовують великомасштабні набори анотованих даних для тренування потужних моделей, які можуть зафіксувати витончені шаблони облич та варіації. Шляхом відображення зображень обличчя у компактні та відрізняючі ознаки представлення, нейронні мережі мають потенціал забезпечити надійні та ефективні системи розпізнавання обличчя.

Основною метою систем розпізнавання обличчя є точна відповідність та ідентифікація осіб на різних зображеннях або відеокадрах. Це включає два ключові етапи: виявлення обличчя, яке знаходиться на зображенні або відеокадрі, та етап верифікації та розпізнавання обличчя, який дозволяє перевірити та ідентифікувати виявлені обличчя з базою даних відомих осіб. Завдяки досягненням у глибинному

навчанні значно поліпшився як процес виявлення облич, так і розпізнавання облич, зробивши їх точнішими та ефективнішими, ніж будь-коли раніше.

Незважаючи на значний прогрес у розпізнаванні облич, декілька викликів досі потребують уваги. Варіації в позах, освітленні, виразах обличчя та віку, а також наявність аксесуарів можуть ускладнювати точність розпізнавання. Крім того, етичні питання, пов'язані з приватністю та потенційними упередженнями в системах розпізнавання облич, потребують ретельного вирішення, щоб забезпечити їх відповідальне та безупинне впровадження.

## **Постановка проблеми та цілі дослідження**

Традиційні методи аутентифікації за допомогою імен користувачів та паролів у комп'ютерних системах піддаються ризику порушення безпеки та створюють незручності для користувачів. Існує потреба у альтернативному методі аутентифікації, який пропонує покращену безпеку, зручність та користувацький досвід. Розпізнавання облич стало перспективним біометричним методом для аутентифікації користувачів. Однак, розробка надійної та ефективної системи авторизації на основі розпізнавання облич поставляє перед нами ряд технічних викликів, які потребують вирішення.

Основна мета цього дослідження - розробка системи, яка дозволяє авторизацію за допомогою розпізнавання облич, і вирішення таких конкретних цілей дослідження:

Дизайн системи: Розробка системи авторизації на основі розпізнавання облич, яка інтегрується безперешкодно з існуючими механізмами авторизації, забезпечуючи покращену безпеку та зручність для користувачів.

Виявлення та відстеження обличчя: Розробка надійних алгоритмів виявлення та відстеження обличчя для точного визначення місцезнаходження та відстеження обличчя користувача в реальному часі, з урахуванням варіацій пози, освітлення та закриття обличчя.

Виділення та представлення ознак: Вивчення технік виділення ознак та розробка компактного та відмінного представлення обличчя для забезпечення ефективного зіставлення та ідентифікації.



# Розділ 1: Що таке нейронні мережі - історія появи, типи, відмінності

## 1.1 Біологічні нейрони

Нейрон - це основний функціональний блок нервової системи, який відповідає за передачу сигналів та обробку інформації. Він складається з соми (тіла нейрона), дендритів та аксону. Сомат виконує обробку сигналів, дендрити отримують вхідні сигнали, аксон передає сигнали.

Нейрони в організмі людини виконують такі функції:

- передачі інформації
- обробка сигналів
- координація дій
- регуляція функцій організму
- формування нейронних шляхів

Нейрони забезпечують людям здатність обробляти вхідну інформацію. Вони не можуть виконувати завдання самостійно. У живому організмі нейрони з'єднані великими "мережами". Дослідження показують, що приблизно сто мільярдів нейронів складають мозок людини.

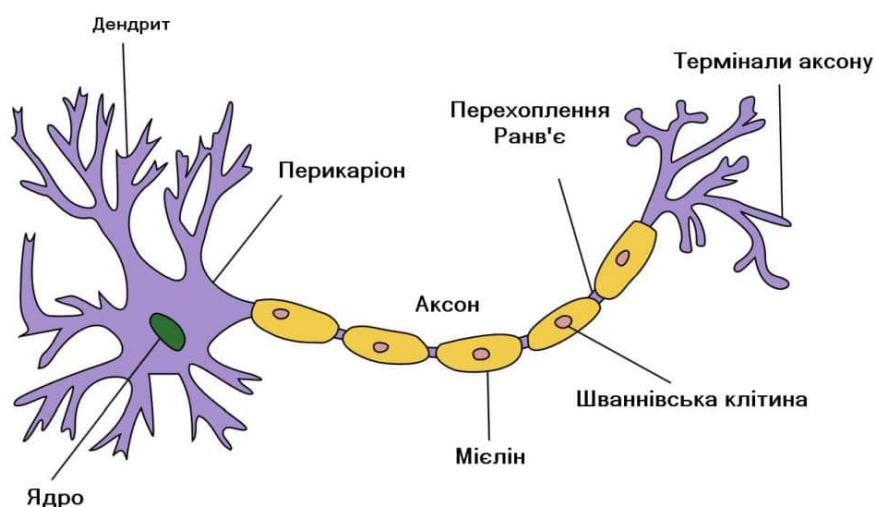


Рис.1.1 Будова нейрона

## 1.2 Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) є математичними моделями, які намагаються відтворити роботу біологічних нейронних мереж. Вони складаються зі з'єднаних штучних нейронів, що обробляють вхідні дані і генерують вихідні сигнали.

При створенні штучних нейронних мереж вчені надихалися біологічною нервовою системою, яка є основою для обробки інформації в мозку. Вони намагалися розібратися в принципах, за якими працюють біологічні нейрони, та застосувати їх у математичних моделях.

У 1943 році Воррен Маккалох (Warren McCulloch) і Уолтер Піттс (Walter Pitts) опублікували роботу "A Logical Calculus of the Ideas Immanent in Nervous Activity", в якій вони запропонували першу математичну модель штучного нейрона. Ця модель базувалася на простих логічних операціях та була важливим кроком у розвитку штучних нейронних мереж. В їх роботі було сформульовано модель нейрона, яка базувалась на простих вхідних сигналах, вагах та пороговій функції активації.

Перші змодельовані нейронні мережі, створені Маккалохом і Піттсом, мали просту структуру і склалися зі штучних нейронів, які виконували логічні операції на вхідних сигналах. Вони були математичними моделями, що допомогли вивчити принципи роботи нейронів і відкрили шлях до подальших досліджень в галузі штучних нейронних мереж.

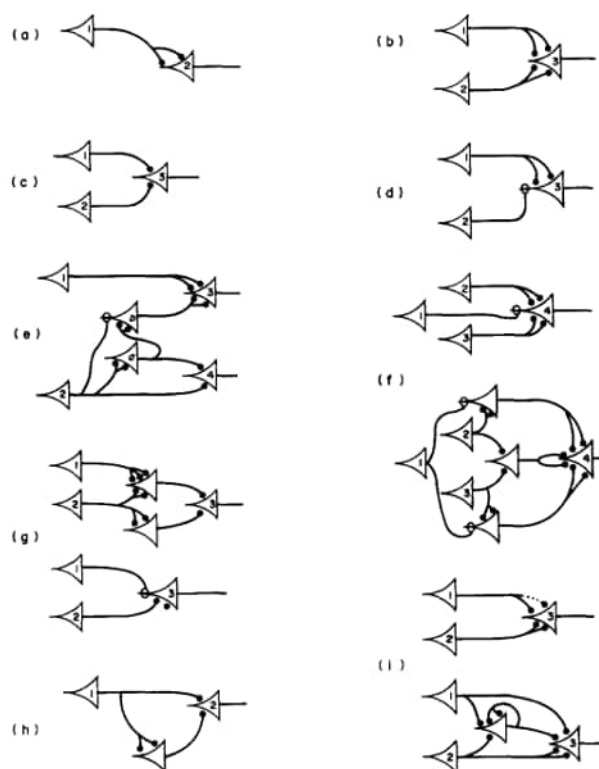


Рис. 1.2 Схема нейронних мереж, запропонованих у статті "A Logical Calculus of the Ideas Immanent in Nervous Activity"

Штучні нейронні мережі (ШНМ) застосовуються в багатьох сферах і вирішують різноманітні завдання. Основні області застосування ШНМ включають машинне навчання, комп'ютерний зір, обробку природних мов, рекомендаційні системи, автономні транспортні засоби та медицину. Завдання, які вони вирішують, включають класифікацію даних, розпізнавання образів, прогнозування, генерацію тексту, розпізнавання мови, рекомендації, виявлення хвороб, аналіз зображень, автономне керування транспортними засобами та багато інших завдань. Застосування ШНМ в цих сферах допомагає автоматизувати процеси, покращувати точність та забезпечувати інтелектуальні рішення у широкому спектрі ділянок людської діяльності.

### 1.3 Будова нейронної мережі

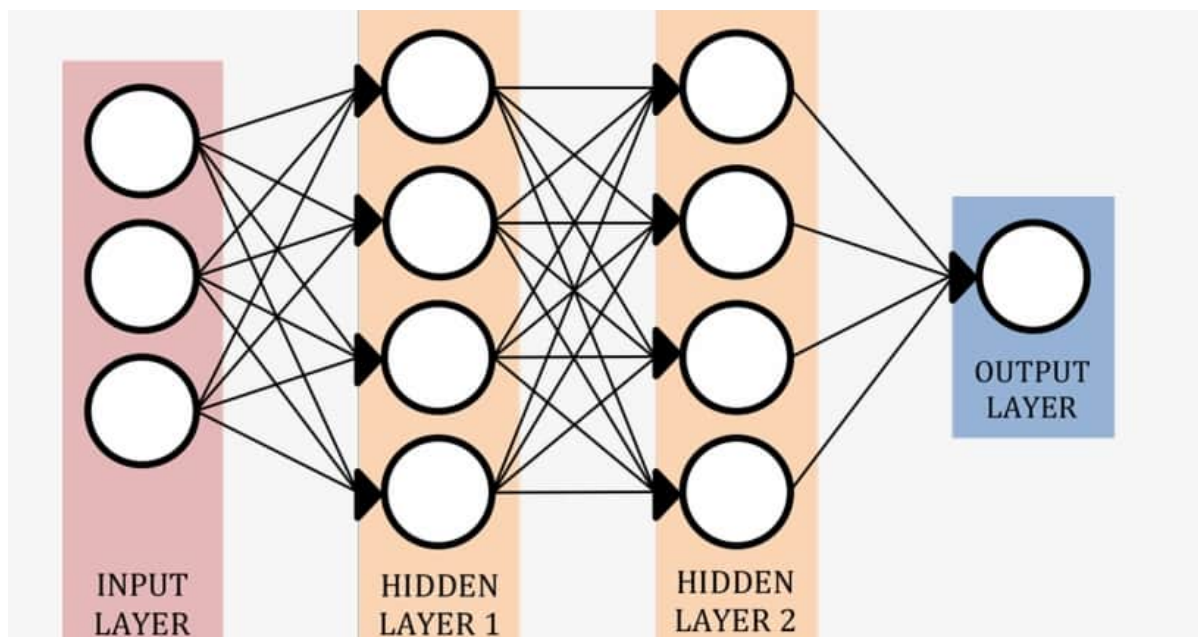


Рис. 1.3 Будова простої нейронної мережі

Проста нейронна мережа з декількома прихованими прошарками, відома як багатошаровий перцептрон, має таку будову:

1. Вхідний прошарок: Це перший прошарок нейронів у мережі, який отримує вхідні дані. Кількість нейронів у цьому прошарку відповідає кількості вхідних ознак або параметрів, які характеризують вхідні дані.

2. Приховані прошарки: Ці прошарки розташовані між вхідним і вихідним прошарками. Кожен прихований прошарок складається з нейронів, які обробляють сигнали від попереднього прошарку і передають їх наступному прошарку. Кількість прошарків і кількість нейронів у кожному прошарку можуть варіюватись залежно від задачі та складності проблеми, яку потрібно вирішити.

3. Вихідний прошарок: Це останній прошарок нейронів у мережі, який генерує вихідні результати або прогнози на основі обробки сигналів в прихованих прошарках. Кількість нейронів у вихідному прошарку залежить від типу завдання, наприклад, для задачі бінарної класифікації може бути один нейрон, а для задачі мультикласової класифікації - декілька нейронів, відповідно до кількості класів.

Вихід кожного прошарку - це також вхід для наступного прошарку. Таким чином, на виході кожен нейрон повертає одне число, а кожен прошарок повертає набір чисел (вектор). Розмірність цього вектору дорівнює кількості нейронів у прошарку.

Кожен нейрон у мережі зв'язаний з нейронами попереднього і наступного прошарків за допомогою зважених з'єднань. З'єднання між нейронами мають ваги, які впливають на сили сигналу, які передаються від одного нейрона до іншого. Ваги навчаються шляхом адаптації під час процесу навчання мережі.

Після того як нейрон приймає вхідний вектор, він виконує з ним три дії:

1. Перемножує його на вектор ваг.
2. Додає значення відхилення до результату множення, так званий bias.
3. Застосовує функцію активації до цього значення.

Таким чином, значення нейрона  $X = \text{activation}(w * i + b)$ , де  $w$  - вектор ваг,  $i$  - вхідний вектор,  $b$  - bias,  $X$  - результат (число),  $\text{activation}$  - функція активації.

Зазвичай ми використовуємо нелінійні функції як функції активації. Це пояснюється тим, що лінійна частина вже оброблена попередньо застосованим добутком і доповненням.

## 1.4 Функції активації

Функції активації є ключовими компонентами нейронних мереж і вони виконують декілька важливих функцій. Основні з них:

1. Нелінійність: Функції активації надають нейронам нелінійність. Це дозволяє нейронним мережам моделювати складні, нелінійні залежності в даних, що робить їх потужними і гнучкими для розв'язання різноманітних задач.

2. Загальне значення: Функції активації перетворюють зважену суму вхідних сигналів в вихідний сигнал з певним діапазоном значень. Це дозволяє нейронам передавати інформацію та виконувати обчислення у визначеному діапазоні значень.

Ось п'ять найважливіших функцій активації:

### 1. Сигмоїдна функція (Sigmoid):

Функція активації сигмоїдного типу перетворює вхідний сигнал в діапазоні від 0 до 1. Вона широко використовується в задачах бінарної класифікації, де потрібно визначити ймовірність належності до певного класу.

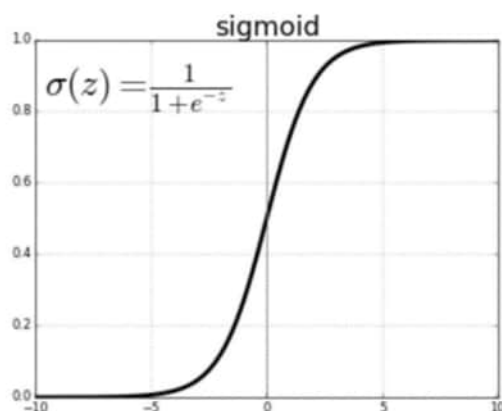


Рис 1.4 Графік сигмоїдної функції

### 2. Гіперболічний тангенс (Tanh):

Функція активації тангенса гіперболічного типу також перетворює вхідний сигнал в діапазоні від -1 до 1. Вона зберігає нелінійність, але може зробити вихід сигналу нульовим при нульовому вхідному сигналі.

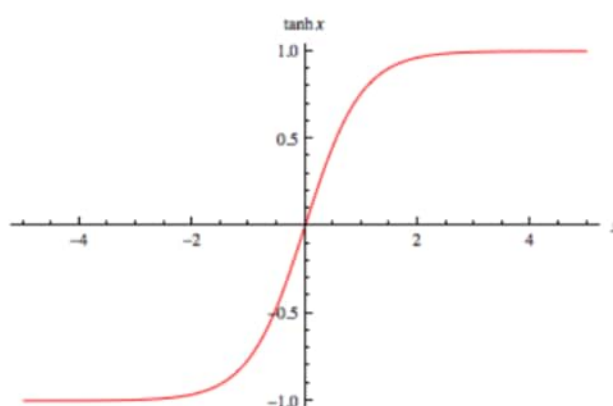


Рис 1.5 Графік функції гіперболічного тангенса

### 3. Rectified Linear Unit (ReLU):

Функція активації ReLU залишає невід'ємні значення вхідного сигналу без змін, а від'ємні значення прирівнює до нуля. Вона швидша у обчисленнях та допомагає уникнути проблеми градієнтного зникнення.

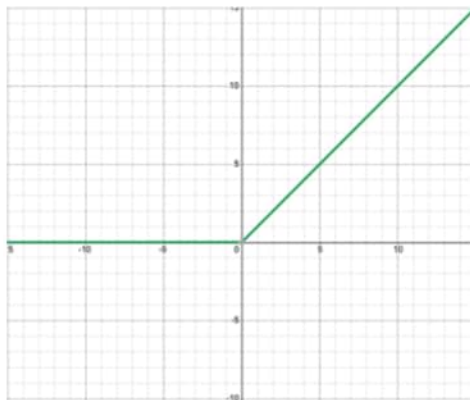


Рис 1.6 Графік функції ReLU

## 1.5 Навчання нейронних мереж

Навчання нейронних мереж може відбуватись з вчителем (навчання з наглядом) або без вчителя (навчання без нагляду). Давайте розглянемо ці два підходи докладніше.

Навчання з вчителем (навчання з наглядом):

У навчанні з вчителем використовується заздалегідь підготовлений набір даних, де кожний приклад має вхідні дані та правильні відповіді. Метою навчання є знайти функцію, яка найкращим чином відображає залежність між вхідними даними та відповідями. Процес навчання полягає в зміні ваг нейронної мережі таким чином, щоб мінімізувати різницю між передбачуваними відповідями та правильними відповідями. Це досягається шляхом використання методів оптимізації та алгоритмів зворотного поширення помилки.

Вихідними даними такої нейронної мережі може бути класифікація чи регресія. Класифікація - визначення, до якої групи належать вхідні дані. Наприклад, розпізнавання літер по картинці або ж розділення картинок з бананами від картинок з апельсинами. Визначення, чи є лист спамом - також одне із застосувань для

нейронних мереж з класифікацією. Регресія - це отримання числового значення, використовуючи вхідні дані. Наприклад, можна визначити скільки має коштувати нерухомість, базуючись на даних з ринку. Також можна спробувати передбачити ціну акцій в майбутньому, спираючись на історичні дані.

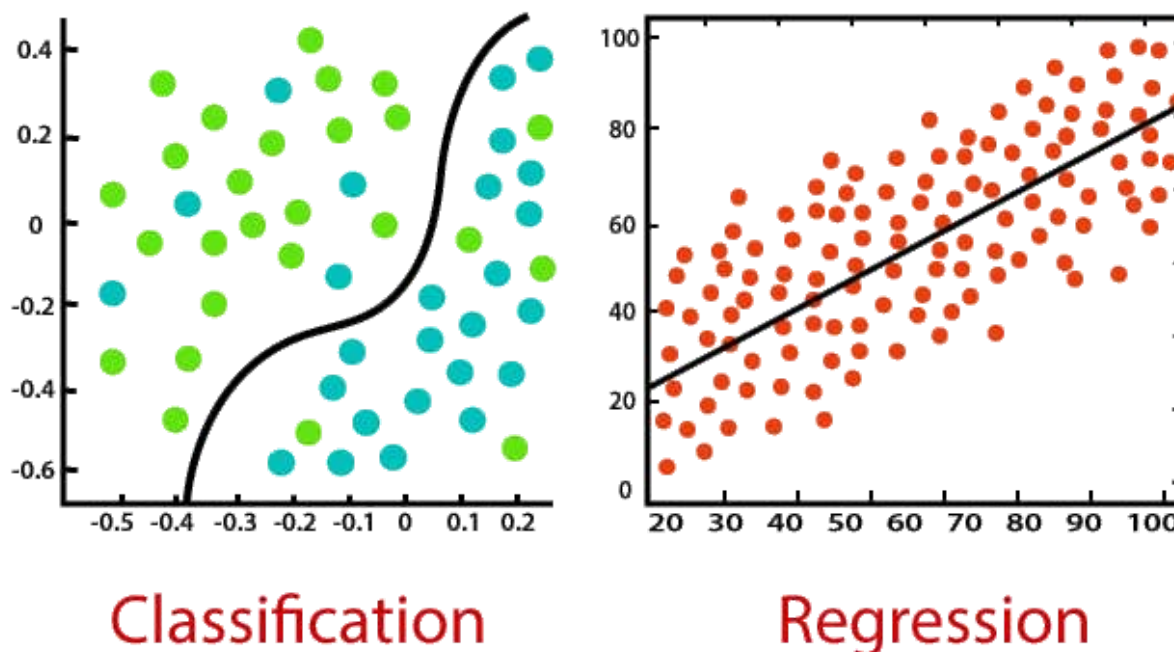


Рис. 1.7 Візуальне представлення вирішень задач класифікації та регресії

Навчання без вчителя (навчання без нагляду):

У навчанні без вчителя немає правильних відповідей, і модель намагається самостійно знайти корисні закономірності та структури у наборі даних. Це може включати кластеризацію даних, знаходження аномалій, зменшення розмірності даних та інші завдання обробки інформації.



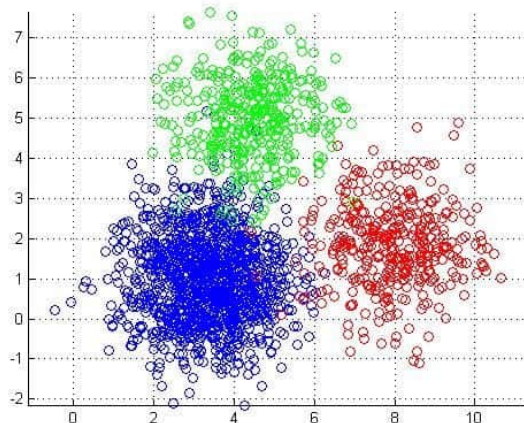


Рис. 1.8 Візуальне представлення вирішення задачі кластеризації

Існують також гібридні підходи, які поєднують елементи навчання з вчителем і без вчителя для досягнення кращих результатів у завданнях навчання мереж. Вибір підходу залежить від характеру даних, задачі та доступності підготовлених наборів даних з правильними відповідями.

## 1.6 Типи нейронних мереж

*Мережа прямого поширення (Feed-forward neural networks).*

Одним із перших типів нейронної мережі був цей. Він називається так через те, що дані рухаються в одному напрямку від входу до виходу. На відміну від RNN, ці мережі не містять циклів.

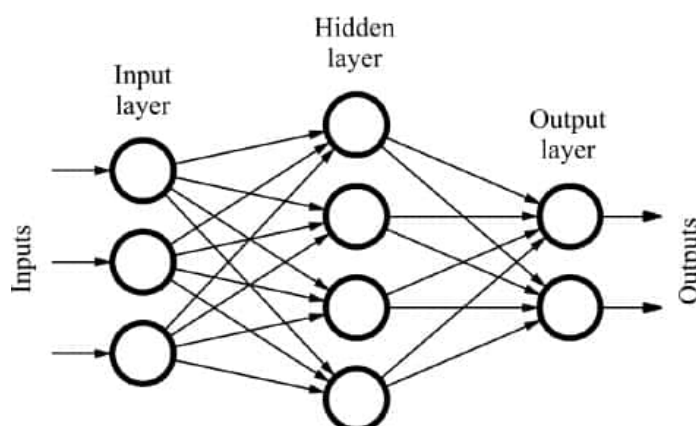


Рис. 1.9 Feed-forward neural network

Данна нейронна мережа використовується в основному під час навчання за допомогою вчителя. У такій мережі кожен нейрон одного прошарку пов'язаний з кожним нейроном наступного прошарку.

Така мережа має перевагу в тому, що вона «не знає» структуру вхідних даних, тобто вона не створена спеціально для обробки зображень чи текстів. Однак її недоліком є те, що вона не працює так добре, як вузьконаправлені нейронні мережі в реальному житті.

### *Рекурентні нейронні мережі (Recurrent neural networks).*

Рекурентні нейронні мережі (RNN) - це тип штучних нейронних мереж, які мають циклічні зв'язки між нейронами, дозволяючи передавати інформацію від попередніх вихідних прошарків до наступних вхідних прошарків. Це робить їх особливо ефективними у розпізнаванні послідовностей та обробці даних, залежних у часі.

Особливістю рекурентних нейронних мереж є наявність пам'яті, що дозволяє їм зберігати і використовувати інформацію з попередніх кроків обробки даних. У кожному прошарку RNN використовується один і той самий набір ваг, які допомагають моделі запам'ятовувати та узагальнювати залежності між елементами послідовності.

У рекурентних нейронних мережах кожен нейрон отримує вхід не тільки з попереднього прошарку, але й свій власний вихід з попереднього кроку часу. Це створює зв'язок між попередніми та поточними станами мережі, що дозволяє аналізувати залежності в часі та прогнозувати наступні значення в послідовності.

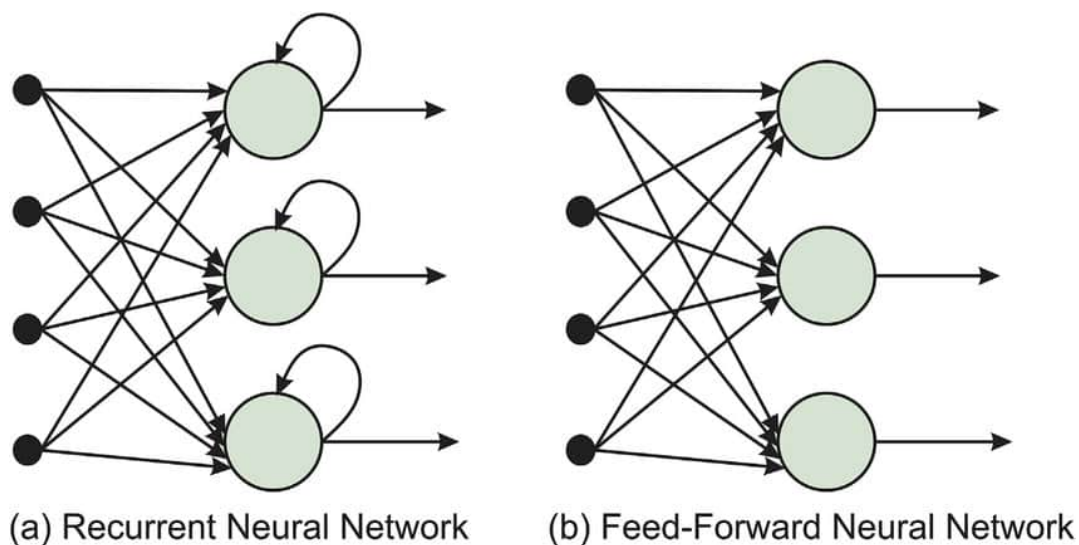


Рис. 1.10 Схематичне порівняння feed-forward нейронних мереж і RNN

У той час як у feed-forward мереж зв'язок вхідних та вихідних даних один до одного, у RNN це може бути один до одного, один до багатьох, багато до одного то багато до багатьох, як показано на Рис 1.11

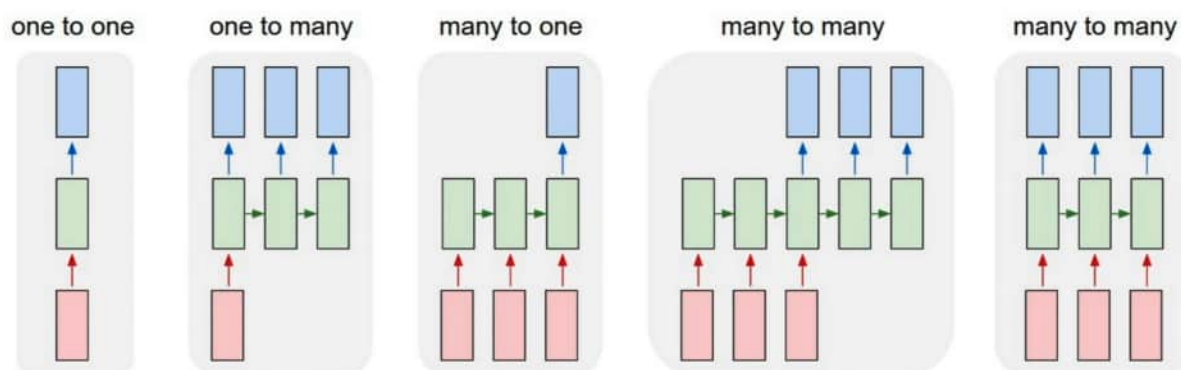


Рис 1.11

*Згорткові нейронні мережі (Convolutional Neural Networks (CNNs)).*

Згорткові нейронні мережі є типом нейронних мереж, спеціально розроблених для ефективної обробки зображень. Перша CNN, відома як LeNet, була запропонована у 1990-х роках. Типова архітектура CNN містить три типи прошарків: convolutional layer, pooling layer та fully connected layer.

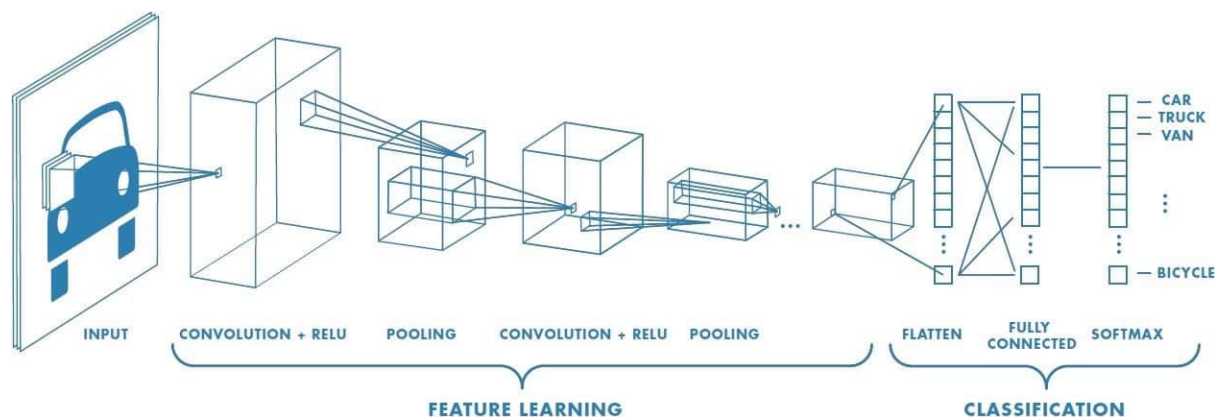


Рис 1.12

Convolutional layer є основним “будівельним матеріалом” в CNN. Ідея полягає в тому, що ці прошарки вміють знаходити певні риси (features) зображення, наприклад лінії, кути, якісь кольорові гамаи тощо. Складність патернів з кожним прошарком може зростати. Це робиться завдяки так званим “фільтрам” (англ. kernel). Фільтри вміють розпізнавати певні патерни(шаблони) в зображеннях. Схематично це показано на рисунку 1.13. Замість літер у просторі реальних чисел будуть числа, які, наприклад, показують RGB значення або значення від 0 до X якщо зображення в форматі grayscale.

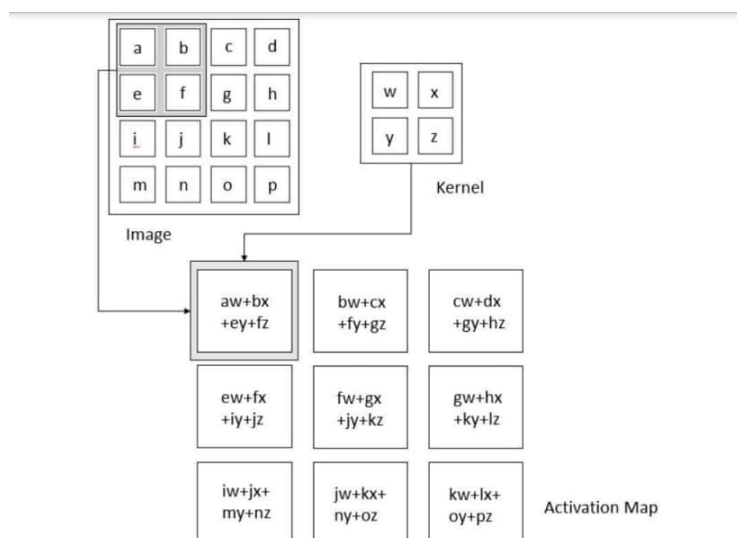


Рис 1.13

Оскільки процес, описаний вище - лінійний, щоб зробити його нелінійним часто додають функцію активації відразу після convolutional layer. Наприклад Sigmoid, TanH, чи ReLU. Pooling прошарки - це прошарки, які отримують на вхід

матрицю, і використовуючи певні правила зменшують її розмір. Це допомагає зменшити кількість параметрів (weights), які мають бути натреновані, тим самим зменшуючи навантаження на обчислювальний пристрій. Також pooling прошарки дозволяють зменшити так зване “перетренування” (англ. overfitting) нейронної мережі. Є maximum pooling, який обирає найбільше значення з підматриці, це показано на Рис 1.14

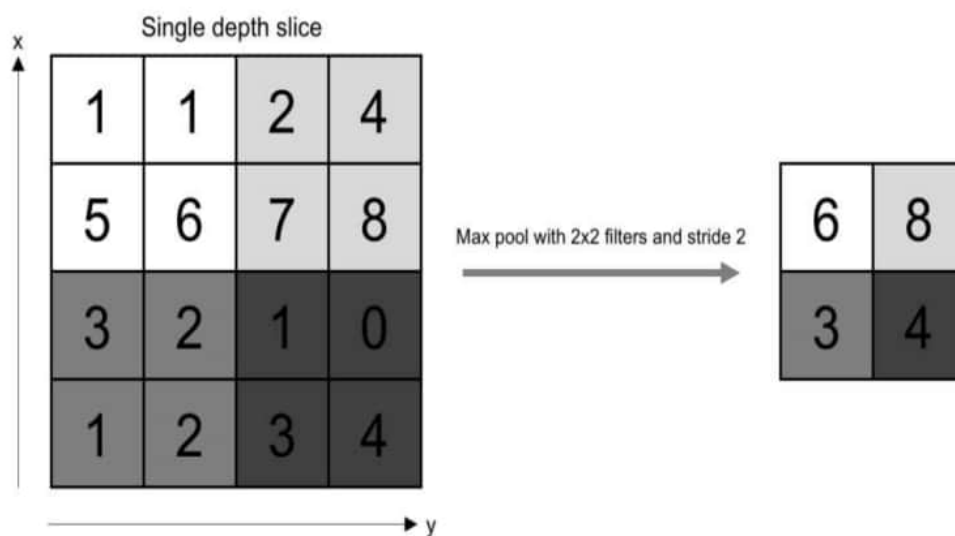


Рис 1.14

Також існує average pooling, який обирає середнє значення для кожної підматриці. Приклад показано на Рис 1.15

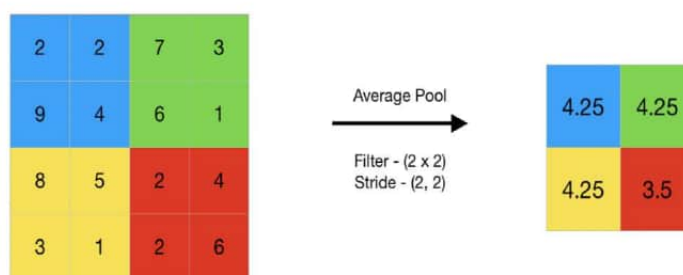


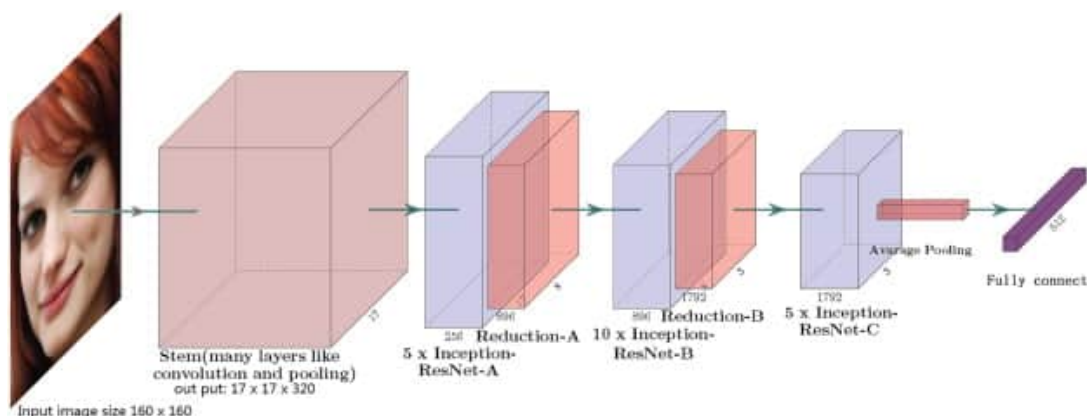
Рис. 1.15

У Fully connected (FC) layers кожен нейрон з одного прошарку пов'язаний з кожним нейроном в попередньому та наступному прошарках. Точно так само, як це відбувалося в feed-forward нейронних мережах. Таким чином, convolutional layers

відповідають за знаходження певних патернів у зображеннях, а FC прошарки відповідають за їхню класифікацію.

## 1.7 Використана модель мережі

На Рис 1.16 показано архітектуру CNN, яка називається Inception ResNet v1. Саме цю архітектуру має модель OpenFace, яку в свою чергу використовує бібліотека `face_recognition`, що була застосована в проекті.



(a) architecture of Inception-ResNet v1

Рис. 1.16

OpenFace використовує архітектуру глибокої нейронної мережі, відому як "Inception ResNet v1". Ця архітектура поєднує ідеї з двох моделей: Inception та ResNet. Вона була спеціально розроблена для задач розпізнавання облич та виявлення особливостей на зображеннях.

Архітектура Inception ResNet v1 складається з глибокого стеку конволюційних прошарків, пулінгу, глобального середнього пулінгу та повнозв'язаних прошарків. Вона має приховані прошарки, які допомагають виявити різноманітні ознаки на зображенні та розпізнавати обличчя.

Кількість прошарків та їх розмір в архітектурі Inception ResNet v1 залежить від конфігурації моделі. Зазвичай вона містить декілька блоків, кожен з яких складається з декількох послідовних прошарків, включаючи згорткові прошарки,

пулінг та повнозв'язані прошарки. Кількість прошарків може сягати декількох десятків.

Загальна кількість прошарків та їх розмір може варіюватись залежно від конфігурації моделі та завдання, для якого використовується OpenFace. Більш глибокі моделі можуть містити більше прошарків для виявлення складніших ознак та розпізнавання більш деталізованих облич.

Загалом, OpenFace використовує глибоку нейронну мережу з архітектурою Inception ResNet v1, яка містить декілька блоків з конволюційними, пулінг та повнозв'язаними прошарками для розпізнавання облич та виявлення особливостей на зображеннях

Для навчання Inception ResNet v1 використовує базу зображень CelebA. CelebA - це набір даних з більш ніж 200 000 зображень обличчя з 10 177 унікальних осіб. Використовується для завдань розпізнавання обличчя, знаходження атрибутів та інших задач обробки зображень.

## Розділ 2: Методологія

### 2.1 Виявлення обличчя, розпізнавання обличчя, верифікація обличчя

Розпізнавання обличчя (face recognition) - це процес ідентифікації або верифікації особи на зображенні або відео шляхом порівняння його/її особливих рис із раніше відомими обличчями. Система розпізнавання обличчя може визначати особу на основі її унікальних характеристик, таких як форма обличчя, розташування очей, носа, рота тощо.

Виявлення обличчя (face detection) - це процес визначення наявності обличчя на зображенні або відео і визначення його положення та рамки. Система виявлення обличчя може виділити та зазначити положення одного або декількох облич у зображенні.

Перевірка обличчя (face verification) - це процес підтвердження, чи належить обличчя певній особі. Вона використовується для перевірки, чи обличчя на зображенні або відео відповідає заданій особі. Під час процесу перевірки система порівнює особливі риси обличчя з даними, що належать до конкретної особи, і визначає, чи вони збігаються.

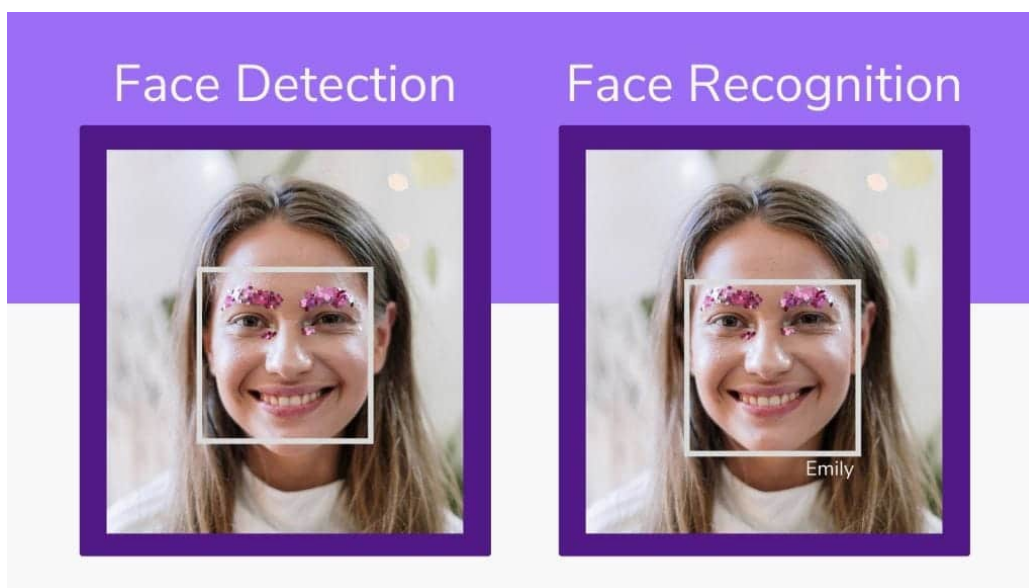


Рис 2.1 Порівняння Face detection vs Face recognition



Отже, розпізнавання обличчя визначає особу, виявлення обличчя визначає його положення, а перевірка обличчя перевіряє відповідність обличчя певній особі.

## 2.2 Огляд методів для виявлення облич, вилучення ознак та класифікації

Процес розпізнавання людини по зображенню обличчя наведено на рис.1 та містить такі кроки :

1. Локалізація обличчя на зображенні (знаходження ділянки зображення, що містить обличчя, її розмірів і положення)
2. Вирівнювання знайденого обличчя (по геометрії і по яскравості)
3. Витяг основних ознак обличчя із зображення
4. Розпізнавання шляхом порівняння вилучених ознак із закладеними у базу даних.

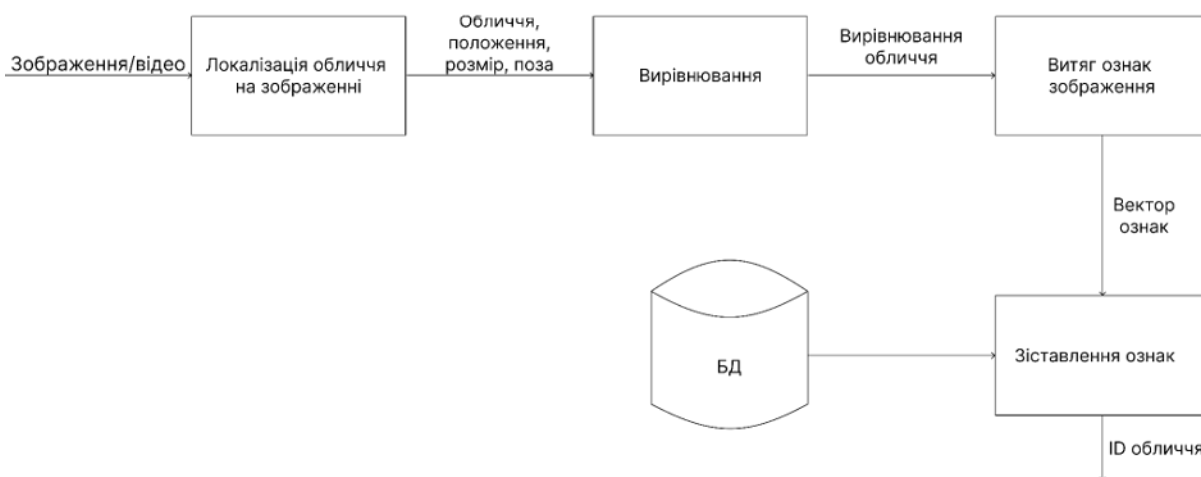


Рис. 1. Загальна схема процесу розпізнавання людини по зображенню обличчя

Ознаки Хаара – ознаки цифрового зображення, використовувані в розпізнаванні образів. Ознаки Хаара використовувалися в першому детекторі облич, що працював в реальному часі . Ознака Хаара складається з суміжних прямокутних

областей. Вони позиціонуються на зображенні, далі сумуються інтенсивності пікселів в областях, після чого обчислюється різниця між сумами. Ця різниця і буде значенням певної ознаки, визначеного розміру, певним чином позиційованої на зображенні. Загальним для всіх зображень є те, що область в районі очей темніше, ніж область в районі щік. Отже загальною ознакою Хаара для обличчя є 2 суміжних прямокутних регіону, що лежать на очах і щоках (рис. 2.).



Рис. 2.2 Ознаки Хаара і приклад їх розташування на обличчі людини

Ключовою особливістю ознак Хаара є найбільша, в порівнянні з іншими ознаками, швидкість. Обчислюється значення такої ознаки за формулою:  $F = X - Y$ , де  $X$  – сума значень яскравості точок закриваються світлою частиною ознаки,  $Y$  – сума значень яскравості точок що закриваються темною частиною ознаки. Ознаки Хаара дають точкове значення перепаду яскравості по осі  $X$  і  $Y$  відповідно. Алгоритм сканування вікна з ознаками виглядає так: обирається вікно сканування та використовувані ознаки для основних рис обличчя; вікно сканування послідовно просувається по зображенню з кроком в 1 клітинку вікна (припустимо, розмір самого вікна 24x24 клітинки); при скануванні зображення в кожному вікні обчислюється приблизно 200 000 варіантів розташування ознак, за рахунок зміни масштабу ознак і їх положення в вікні сканування; сканування проводиться послідовно для різних масштабів скануючого вікна; всі знайдені ознаки потрапляють до класифікатору, який фіксує наявність рота, носа, та іншої ознаки певної розмірності у певному регіоні і знаходить найближчий зразок у базі.

Класифікатор повинен реагувати тільки на певну, підмножину всіх ознак, якою він навчається з навчальної вибірки. У якості класифікатора може бути перцепторн, ієрархічні алгоритми та інші алгоритми класифікації. Вхідні дані: розмір та розташування очей, розмір та розташування рота, розмір та розташування інших головних рис.

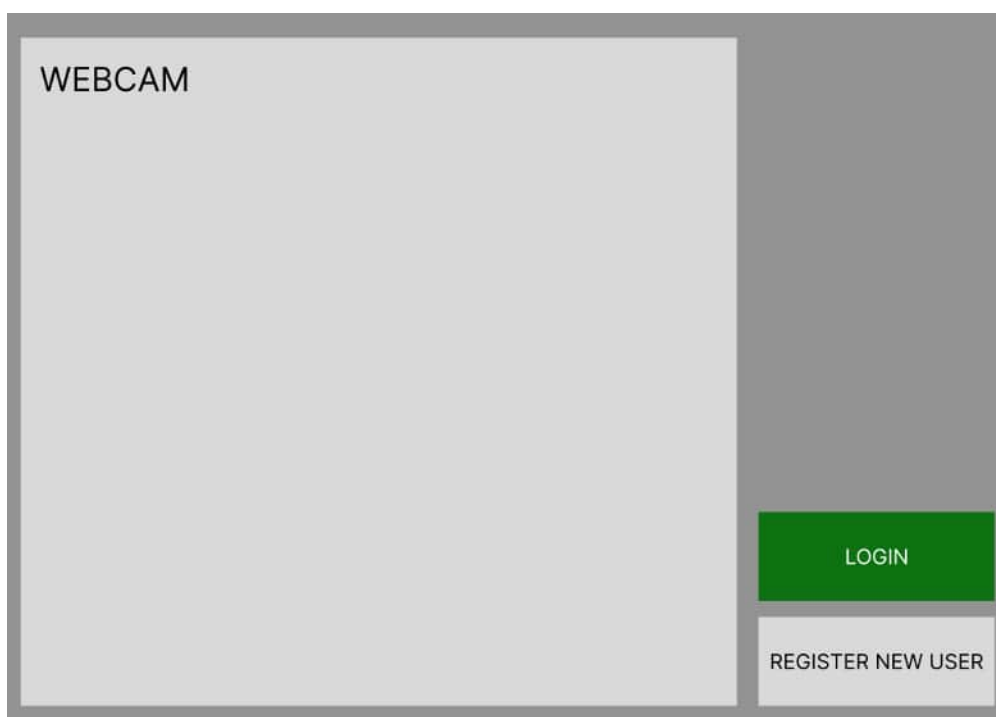
Для поліпшення точності використовується технологія бустинга. Бустинг – комплекс методів, що сприяють підвищенню точності аналітичних моделей. В його основі лежить побудова каскада класифікаторів, кожен з яких (крім першого) навчається на помилках попереднього. Наприклад, один з перших алгоритмів бустинга використовував каскад з 3-х моделей, перша з яких навчалася на всьому наборі даних, друга – на вибірці прикладів, в половині з яких перша дала правильні відповіді, а третя – на прикладах, де «відповіді» перших двох розійшлися. Таким чином, має місце послідовна обробка прикладів каскадом класифікаторів, причому так, що завдання для кожного наступного стає важче. Результат визначається шляхом простого голосування: приклад відноситься до того класу, який виданий більшістю моделей каскаду.

## Розділ 3: Реалізація

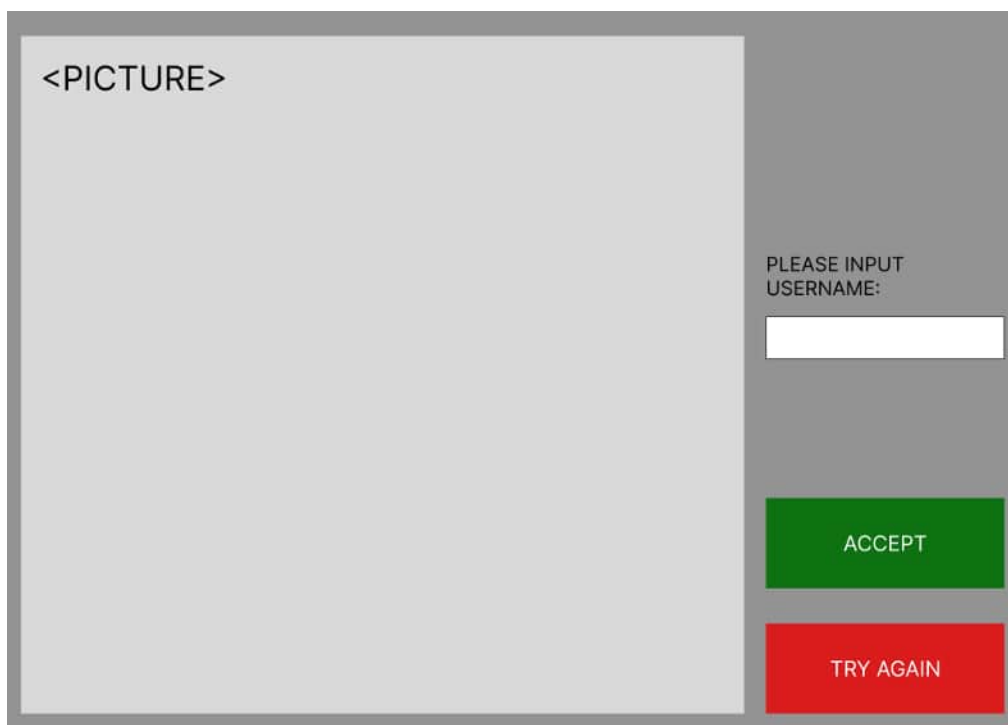
### 3.1 Розробка UI/UX дизайну в середовищі Figma.

Варто відзначити, що важливо було створити окрему одиницю програмного забезпечення, яку можна інтегрувати для різних систем і на різних платформах. Важливо було зберегти простий і зрозумілий для користувача інтерфейс, зробити використання аутентифікації за допомогою розпізнавання облич швидким та зрозумілим.

Для цього я створила 5 мокапів, які вміщують в себе функціонал проекту і можливі виходи з системи.

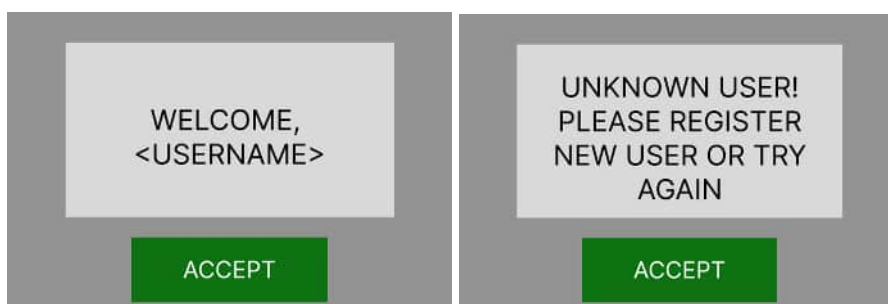


Головний екран дає чітке розуміння, що при вході є дві опції - логін та реєстрація нового користувача.



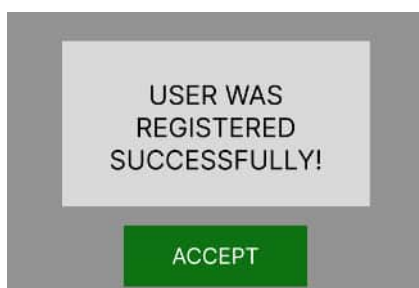
The image shows a registration screen layout. On the left, there is a large grey rectangular area labeled "<PICTURE>". On the right, there is a vertical column of elements: the text "PLEASE INPUT USERNAME:" above a white input field, a green button labeled "ACCEPT", and a red button labeled "TRY AGAIN".

Екран з реєстрацією нового користувача також має зрозумілий та простий інтерфейс, дає на 2 можливих варіанти розвитку подій.



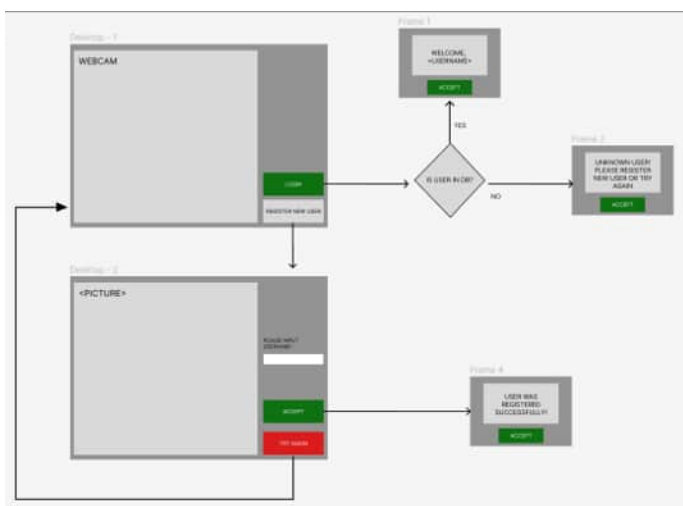
The image shows two possible outcomes of the registration process, each in a grey frame. The left frame contains a grey box with the text "WELCOME, <USERNAME>" and a green button labeled "ACCEPT". The right frame contains a grey box with the text "UNKNOWN USER! PLEASE REGISTER NEW USER OR TRY AGAIN" and a green button labeled "ACCEPT".

Вибравши на головному екрані опцію "LOGIN", операція може мати 2 виходи і це передбачено програмою.



The image shows a successful registration message in a grey frame. It contains a grey box with the text "USER WAS REGISTERED SUCCESSFULLY!" and a green button labeled "ACCEPT".

При успішній реєстрації користувача отримуємо повідомлення про успіх.



Також можемо побачити схему роботи всієї системи.

## 3.2 Використані технології

Python та бібліотеки глибокого навчання. Python - популярна мова програмування, відома своєю простотою, зрозумілістю та широким спектром бібліотек і фреймворків. Вона здобула значну популярність в галузі глибокого навчання та штучного інтелекту завдяки своєму багатому екосистемі та підтримці спільноти. Python надає універсальне та гнучке середовище для розробки моделей глибокого навчання, зокрема для розпізнавання облич у системі авторизації. У цьому розділі ми обговоримо відповідність та переваги Python, а також деякі поширені бібліотеки глибокого навчання.

Відповідність Python:

Python широко використовується у галузі науки про дані та машинного навчання, що робить його природним вибором для впровадження моделей глибокого навчання. Простота його використання роблять Python доступним для дослідників, розробників та ентузіастів. Багата екосистема бібліотек Python надає широкий спектр інструментів та фреймворків, які спрощують процес розробки та спрощують складні завдання.

Переваги Python для глибокого навчання:

1. Простота та зрозумілість: Чистий та зрозумілий синтаксис Python полегшує розуміння та підтримку коду, зменшуючи криву навчання для розробників. Він сприяє ефективній співпраці та спільному використанню коду серед дослідників та практиків.

2. Велика кількість бібліотек: Python пропонує безліч бібліотек, спеціально розроблених для завдань глибокого навчання. Деякі популярні бібліотеки включають TensorFlow, PyTorch, Keras та Theano. Ці бібліотеки надають високорівневі абстракції, готові прошарки нейронних мереж та алгоритми оптимізації, спрощуючи реалізацію складних архітектур нейронних мереж.

3. Підтримка спільноти: У Python є велика та активна спільнота розробників, дослідників та ентузіастів, які сприяють розвитку та підтримці бібліотек глибокого навчання. Ця жвава спільнота забезпечує постійне вдосконалення, виправлення помилок та багатство онлайн-ресурсів, навчальних посібників та форумів для підтримки.

4. Можливості інтеграції: Python безпроблемно інтегрується з іншими мовами та технологіями, що дозволяє легко поєднувати його з існуючими системами, базами даних, веб-фреймворками або API. Це сприяє інтеграції розробленої системи авторизації на основі розпізнавання облич з іншими компонентами або програмами.

Бібліотеки глибокого навчання:

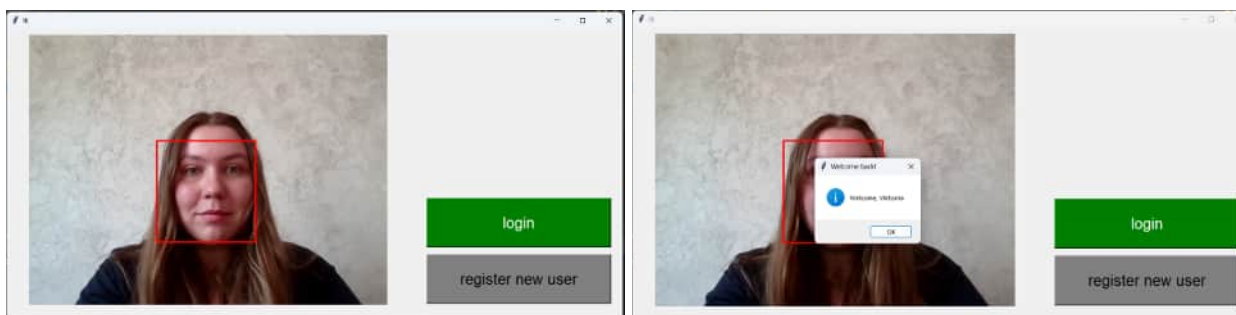
У Python існує кілька бібліотек глибокого навчання, які надають комплексні інструменти та фреймворки для реалізації систем розпізнавання облич. Ці бібліотеки надають широкий спектр функціональності, включаючи готові моделі, утиліти передобробки зображень, алгоритми оптимізації та інструменти візуалізації, що значно спрощують процес розробки системи авторизації на основі розпізнавання облич. В моїй роботі я використала бібліотеку `face_recognition`.

### 3.3 Опис роботи

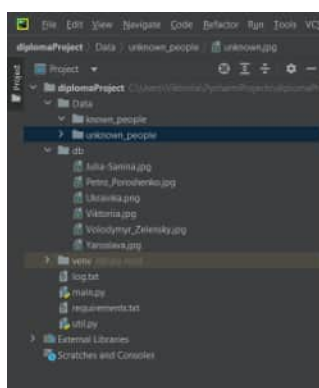
Моя система обліку присутності облич є надійним та універсальним рішенням,

розробленим з використанням мови програмування Python. Вона використовує потужність розпізнавання облич для інновації процесу відстеження присутності.

У центрі системи знаходиться механізм розпізнавання обличчя який застосовується для входу в систему і гарантує доступ лише авторизованим особам. Цей механізм входу є заходом безпеки, захищаючи дані про присутність і запобігаючи несанкціонованому доступу.



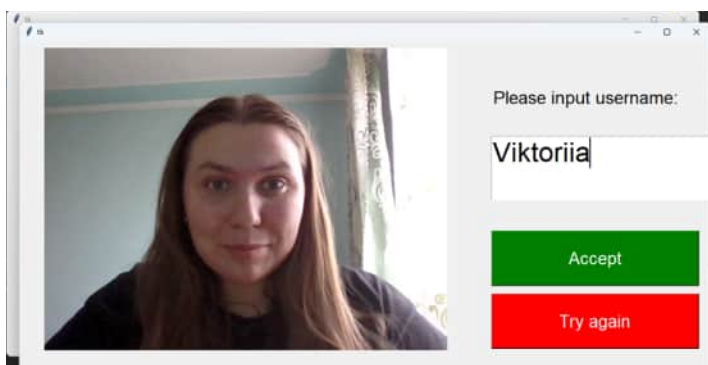
Для того щоб зрозуміти краще, поглянемо на структуру проекту



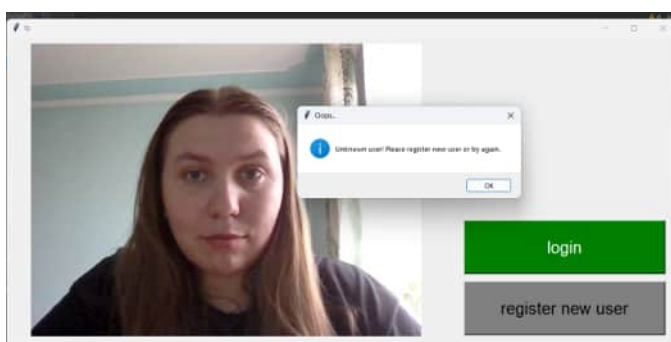
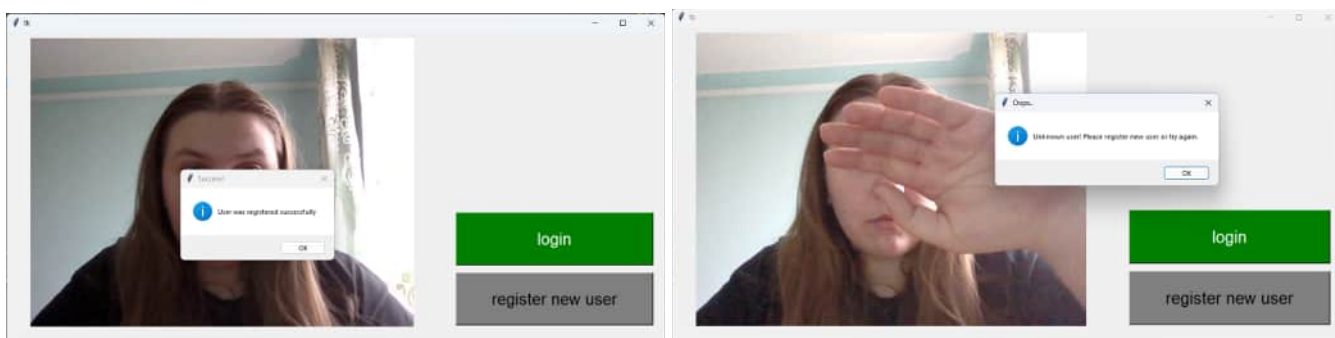
Бачимо папку “db”, саме там система проводить пошук для розпізнавання обличчя користувача. В цей момент вона застосовує ознаки Хаара для розпізнавання обличчя.

Важливим аспектом системи обліку присутності облич є її здатність реєструвати нових користувачів. Адміністратори або уповноважений персонал можуть використовувати цю функцію для реєстрації осіб у системі, захоплюючи їх обличчя. Цей процес передбачає використання потужних можливостей виявлення та розпізнавання облич бібліотеки cv2 (OpenCV). Шляхом точного захоплення та пов'язування зображень облич з іменами користувачів система створює основу для точної ідентифікації під час наступних процесів обліку присутності.

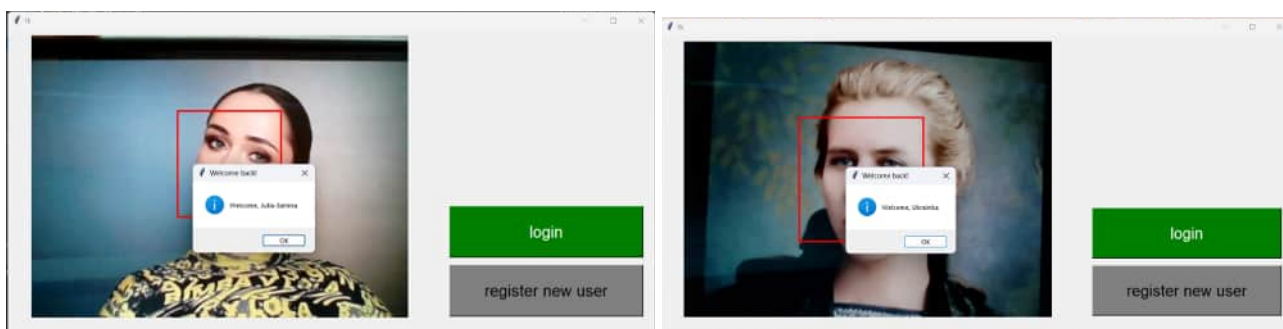




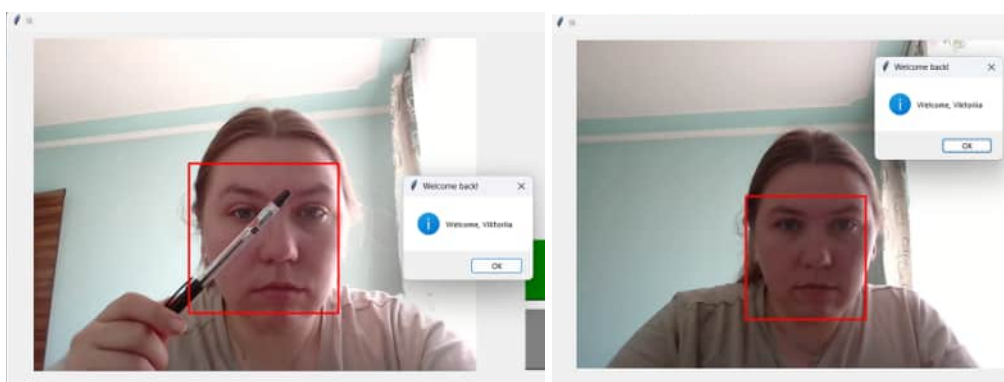
Система обліку присутності облич надає перевагу користувачам, комбінуючи візуальні та текстові елементи у своєму графічному інтерфейсі користувача. Зрозумілі візуальні підказки дозволяють користувачам легко навігувати по системі та з легкістю виконувати дії. Крім того, система надає інформативні повідомлення про успіх або помилки, щоб користувачі вчасно були обізнані про стан системи та можливі проблеми.



Перевіримо також вхід інших відомих системі користувачів.



### Оцінка якості роботи мережі.



Було протестовано мережу на стійкість до шумів. Загалом оцінка точності розпізнавання мережі складає 80%. Мережа достатньо стійка до шумів.

Для забезпечення ефективного керування файлами та каталогами система використовує бібліотеку `os.path`.

```

Viktoriia,2023-05-13 11:13:04.857797
Viktoriia,2023-05-13 11:16:55.857797
Viktoriia,2023-05-28 16:18:05.631129
Viktoriia,2023-05-28 16:19:01.631129
Viktoriia,2023-04-11 12:13:35.998238
Viktoriia,2023-04-11 12:14:48.998238
Viktoriia,2023-06-13 13:11:08.786567
Viktoriia,2023-06-14 19:22:59.947219
Viktoriia,2023-06-16 12:52:42.773496
Viktoriia,2023-06-16 12:54:34.773496
11

```

Ця універсальна бібліотека надає функції для маніпулювання шляхами, дозволяючи системі організувати та зберігати зображення облич та записи присутності у структурованому вигляді. Шляхом ефективної обробки файлів система забезпечує цілісність та доступність важливих даних.

Бібліотека PIL (Python Imaging Library) відіграє важливу роль у обробці та маніпуляції зображень в системі обліку присутності облич. Ця бібліотека оснащує систему різноманітними інструментами та функціями для завдань, таких як зміна розміру, обрізка та покращення зображень. Ці можливості дозволяють оптимізувати зображення облич, забезпечуючи сталу якість та точні результати розпізнавання.

Для подальшого розширення можливостей системи використовується бібліотека subprocess для виконання зовнішніх команд. Ця функціональність дозволяє безперервну інтеграцію з зовнішніми додатками або системами, сприяючи завданням, таким як синхронізація даних, генерація звітів або взаємодія з іншими модулями або API. Здатність виконувати зовнішні команди відкриває можливості для розширення функціональності системи згідно з конкретними вимогами.

Маркування присутності в системі є безшовним процесом, який забезпечується можливостями розпізнавання облич бібліотеки cv2. В реальному часі система виявляє обличчя та порівнює їх з зареєстрованими обличчями, захопленими під час реєстрації. Коли виявляється відповідність, система реєструє присутність, фіксуючи дату та час за допомогою бібліотеки datetime. Цей точний та автоматизований процес усуває необхідність вручного маркування присутності, що зберігає час і зменшує ймовірність помилок.

Записи про присутність, включаючи імена користувачів та відповідні часи входу, зберігаються в окремому текстовому файлі. Цей підхід забезпечує простий та ефективний спосіб зберігання та отримання даних про присутність. Однак для реалізації масштабних рішень або в ситуаціях, де потрібне інтенсивне управління даними, систему можна розширити для використання спеціалізованої системи управління базами даних. Інтеграція з базою даних надає переваги, такі як масштабованість, ефективні запити та надійні можливості управління даними.

Орієнтуючись на майбутнє, система обліку присутності облич є надійною основою для подальшого розвитку та вдосконалення. Одним з потенційних напрямків росту є впровадження звітів про присутність та статистики, які надають користувачам та адміністраторам зрозумілу та інформативну інформацію про присутність в певні періоди часу. Це може бути корисно для вимірювання продуктивності, контролю робочого графіку та аналізу показників присутності.

Завжди є можливість продовжувати розробку та вдосконалення системи, використовуючи інструменти та бібліотеки, які вже використовуються, або додавши нові функції та інтеграції з іншими системами залежно від вашого специфічного випадку використання.

## Висновки

Під час виконання дипломної роботи були розв'язані наступні задачі:

- проведено розгляд сучасних програмних комплексів для розпізнавання образів;
- проведено розгляд застосування нейромереж в розпізнаванні образів;
- проведено розгляд типів нейронних мереж;
- проведено розгляд підходів до навчання нейронних мереж;
- проведено аналіз вимог до програмного та апаратного забезпечення розробленого програмного модуля;
- проведено вибір середовища розробки та технологій, застосованих в розробленому програмному модулі;
- розроблено програмний код в *Pycharm Community*;
- протестовано програмний модуль на зображеннях та відеофайлах.

У дипломній роботі досліджено поняття нейронна мережа, глибоке навчання, згортова нейронна мережа, інтегроване середовище розробки.

Проведено огляд нейромереж різних типів: нейронних мереж прямого поширення, багат шарового процетрона, згорткової нейронної мережі, рекурентної нейронної мережі, модульних нейронних мереж та способів їх навчання.

На основі розглянутих моделей нейромереж, було обрано згорткову нейромережу Inception ResNet v. Вона дозволяє вилучати різноманітні ознаки з зображень, а завдяки глибокому навчанню і великим наборам даних(CelebA), ця модель може досягати високої точності і надійності в розпізнаванні облич. Програмний продукт був реалізований у середовищі візуального програмування *PyCharm Community*.

Система відстеження присутності з використанням розпізнавання обличчя, розроблена в даній роботі, спрощує та автоматизує процеси відстеження присутності. Точність, ефективність, безпека та зручний інтерфейс користувача сприяють покращенню моніторингу відвідуваності в різних сферах, включаючи

освітні установи, робочі місця та інші організації. Завдяки використанню технології розпізнавання обличчя та впровадженню вдосконалених технік, ця система створює передумови для подальшого розвитку відстеження присутності та пов'язаних застосувань.

## Список використаної літератури

1. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Заповнення прогалини у розпізнаванні облич у рівність з досягненнями людини. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 1701-1708.
2. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Розпізнавання облич з використанням глибинного навчання. Британська конференція з машинного бачення, 41.1-41.12.
3. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: Уніфіковане вбудовування для розпізнавання та кластеризації облич. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 815-823.
4. Sun, Y., Wang, X., & Tang, X. (2014). Глибинне навчання представлення облич за допомогою спільної ідентифікації-перевірки. Переваги в нейроінформатиці, 1988-1996.
5. Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). Дискримінантний підхід до навчання ознак для глибокого розпізнавання облич. Європейська конференція з комп'ютерного зору, 499-515.
6. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). SphereFace: Глибоке вбудовування гіперсфери для розпізнавання облич. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 212-220.

7. Deng, J., Guo, J., & Zafeiriou, S. (2019). ArcFace: Додаткова кутова втрата для глибокого розпізнавання облич. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 4690-4699.

8. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Щільно з'єднані згорткові мережі. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 4700-4708.

9. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Глибоке залишкове навчання для розпізнавання зображень. Доповіді конференції IEEE з комп'ютерного зору та розпізнавання образів, 770-778.

10. Simonyan, K., & Zisserman, A. (2014). Дуже глибокі згорткові мережі для розпізнавання великомасштабних зображень. Препринт arXiv arXiv: 1409.1556.

11. Why do Neural Networks Need an Activation Function? -

<https://towardsdatascience.com/why-do-neural-networks-need-an-activation-function-3a5f6a5f00a>