МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
В.о. завідувача кафедри інтелектуальних інформаційних систем, канд. техн. наук, доц.
_____Є. В. Сіденко
«____»_____202_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СИНТЕЗ ТА ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ СПІВРОБІТНИКІВ ПІДПРИЄМСТВА

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21710124

Виконав студент 6-го курсу, групи 601
_____*Проворний О.В.*____
(підпис, ініціали та прізвище)
«___» _____ 2023 р.

Керівник: доцент кафедри, к.т.н., доцент
(наук. ступінь, вчене звання)
_____Козлов О. В.___
(підпис, ініціали та прізвище)
«___» _____ 2023 р.

Миколаїв – 2023

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри інтелектуальних інформаційних систем, канд. техн. наук, доцент
_____ Є. В. Сіденко
«____» _____ 20 р.

ЗАВДАННЯ
на магістерську кваліфікаційну роботу
Проворному Олегу Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Синтез та дослідження нейромережевої системи для розпізнавання облич співробітників підприємства».
Керівник роботи Козлов Олексій Валерійович, доцент кафедри, к.т.н., доцент.
Затв. наказом Ректора ЧНУ ім. Петра Могили від «03» листопада 2022 р. № 199
2. Строк подання студентом роботи 16 лютого 2023 р.
3. Вхідні (початкові) дані до роботи: існуючі системи для розпізнавання облич, набори даних, існуючі моделі нейронних мереж з розпізнавання облич. Очікуваний результат: система контролю та управління доступом для підприємства.
4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):
   - аналіз існуючих на ринку видів інтелектуальних систем розпізнавання облич;
   - аналіз основних видів інформаційних моделей штучних нейронних мереж.
   - вибір оптимальної інформаційної моделі нейронної мережі для вирішення завдання;
   - визначення основних інструментальних засобів для розробки системи;

– проєктування системи розпізнавання облич співробітників підприємства.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: розглянути питання охорони праці на робочому місці в офісі компанії, виконати інтегральну оцінка умов праці та запропоновати заходи, спрямовані на їх покращення.

7. Консультанти:

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис**
Спеціальна частина з охорони праці		
Методична частина		

Керівник роботи __ доцент кафедри, к.т.н., доцент Козлов О. В._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Проворний О. В._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «__07__» _____листопада____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН
Виконання магістерської кваліфікаційної роботи

Тема: Синтез та дослідження нейромережевої системи для розпізнавання облич співробітників підприємства

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2022	20.10.2022	Виконано
2	Отримання завдання на виконання МКР	21.10.2022	10.11.2022	Виконано
3	Складання календарного плану на період виконання МКР	11.11.2022	15.11.2022	Виконано
4	Огляд літератури за темою дослідження	16.11.2022	27.11.2022	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	28.11.2022	18.12.2022	Виконано
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	19.12.2022	12.01.2023	Виконано
7	Опис фахової частини МКР, зокрема дослідження публікацій щодо розпізнавання облич, огляд існуючих архітектур штучних нейронних мереж для вирішення поставленої задачі, реалізація обраних технологій з аналізом отриманих результатів	13.01.2023	25.01.2023	Виконано
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2023	02.02.2023	Виконано
9	Попередній захист МКР на засіданні комісії кафедри	03.02.2023	03.02.2023	Виконано
10	Корегування роботи за результатами попереднього захисту	04.02.2023	06.02.2023	Виконано
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	07.02.2023	09.02.2023	Виконано
12	Подання МКР рецензенту	09.02.2023	10.02.2023	Виконано
13	Рецензування МКР	11.02.2023	12.02.2023	Виконано
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.02.2023	16.02.2023	Виконано
15	Захист МКР перед екзаменаційною комісією (ЕК)	22.02.2023	23.02.2023	Виконано

Розробив студент Проворний О. В. _____
(прізвище та ініціали) (*підпис*)

Керівник роботи доцент кафедри, к.т.н., доцент Козлов О. В. _____
(*наук. ступінь, вчене звання, прізвище та ініціали)* *(підпис)*

«12» листопада 2022 р.

АНОТАЦІЯ

магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Проворного Олега Вікторовича

на тему: **«СИНТЕЗ ТА ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ СПІВРОБІТНИКІВ ПІДПРИЄМСТВА»**

Актуальність магістерської кваліфікаційної роботи полягає у використанні нейронних мереж для вирішення погано формалізованих завдань в інтелектуальному аналізі даних. Зростання обсягів інформації, а також розширення кола технічно складних завдань прийняття рішень вимагають систематизації існуючих методів та розробки нових методик та алгоритмів розв'язання.

Об'єктом дослідження є процеси розпізнавання облич за допомогою нейромережевих систем.

Предметом дослідження є методи, моделі та програмні засоби розпізнавання облич за допомогою штучних нейронних мереж.

Метою даної магістерської кваліфікаційної роботи – є підвищення ефективності системи контролю доступу співробітників підприємства за допомогою використання штучних нейронних мереж для розпізнавання облич.

У першому розділі розглянуті загальні питання штучних нейронних мереж, проаналізовані наявні системи розпізнавання облич.

У другому розділі проведено аналіз та вибір інструментальних засобів розробки системи розпізнавання облич. Було проведено загальний аналіз та розглянуто процеси машинного навчання.

У третьому розділі представлена реалізація та розробка системи для розпізнавання облич співробітників підприємства.

Магістерська кваліфікаційна робота містить 95 сторінки, 48 рисунків, 4 таблиць, 35 використаних джерела та 2 додатки.

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Provornyi Oleh

"SYNTHESIS AND RESEARCH OF A NEURAL NETWORK SYSTEM FOR RECOGNIZING THE FACES OF COMPANY EMPLOYEES"

The relevance of the master's qualification work is the use of neural networks to solve poorly formalized tasks in intelligent data analysis. The increase in the amount of information, as well as the expansion of the range of technically complex decision-making tasks, require the systematization of existing methods and the development of new methods and solving algorithms.

The object of research is the process of face recognition using neural network systems.

The subject of research are methods, models and software tools for face recognition using artificial neural networks.

The purpose of this master's thesis is to improve the efficiency of the access control system of the company's employees using artificial neural networks for face recognition.

In the first chapter, general issues of artificial neural networks are considered, existing face recognition systems are analyzed.

In the second section, the analysis and selection of tools for the development of the face recognition system were carried out. A general analysis was conducted and machine learning processes were considered.

The third chapter presents the implementation and development of a system for recognizing the faces of company employees.

Master's thesis contains 95 pages, 48 figures, 4 tables, 35 used sources and 2 appendices.

ЗМІСТ

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«СИНТЕЗ ТА ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ СПІВРОБІТНИКІВ ПІДПРИЄМСТВА»

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21710124

Виконав студент 6-го курсу, групи 601
_____*О. В. Проворний*____
(підпис, ініціали та прізвище)
«___» _____ 2023 р.

Керівник: _доцент кафедри, к.т.н., доцент_
(наук. ступінь, вчене звання)
_____*О. В. Козлов*____
(підпис, ініціали та прізвище)
«___» _____ 2023 р.

Миколаїв – 2023

ВСТУП

У сучасному світі інтелектуальні системи мають значну популярність через те що мають великі можливості та ефективності використання.

Існує багато завдань, таких як: робототехніка, економіка та бізнес, фізика, математика, охоронні системи та безпека, медицина та багато інших для яких потрібен практичний дозвіл з використанням інтелектуальних систем. Такий великий обсяг областей застосування, може сказати про те, що інтелектуальні системи є унікальним набором для вирішення завдань з аналізу та обробки великих обсягів даних, вирішення завдань різних рівнів складності.

Нейронна мережа – це структура для обчислювання, що складається з багатьох складових одного типу. Ці складові потрібні для виконання простих функцій, а всі процеси, які відбуваються у штучній нейронній мережі, можна порівняти з процесами, які відбуваються в нервових системах живих організмів.

У нейронних мережах відсутня явна залежність, тому що вони нелінійні за своєю природою, що дозволяє одночасно використовувати розроблену технологію. Лінійне моделювання було основним методом моделювання впродовж багатьох років, оскільки для нього були добре розроблені процедури оптимізації.

Актуальність магістерської кваліфікаційної роботи полягає у тому, щоб за допомогою нейронних мереж вирішувати погано формалізовані завдання в інтелектуальному аналізі даних. Збільшення кількості інформації, а також збільшення обсягу технічно складних задач прийняття рішень вимагають розробки нових методик та алгоритмів розв'язання, систематизації існуючих методів. Штучні нейронні мережі надають перспективи у розвитку, а від їх використання програмне забезпечення отримує велику перевагу. Крім того, кожне завдання, що реалізується, має необмежений і нестандартний набір методів рішення. У магістерській кваліфікаційній роботі розглядається можливість застосування нейронних мереж для вирішення задачі розпізнавання обличчя людини.

На сьогоднішній день більшість підприємств уже почали використовувати у своїх охоронних системах системи контролю та управління доступом (СКУД) [1-3]. Адже завдяки таким біометричним системам ідентифікації можна значно підвищити безпеку підприємства та його працівників.

Раніше для розпізнавання людей на прохідних підприємства встановлювали електронні турнікети зі зчитувачами карт або відбитків пальців, але сьогодні, у зв'язку з бурхливим розвитком біометричних технологій, компанії все частіше переходять на інші методи розпізнавання, точніші та зручніші. Наприклад, розпізнавання облич по відеопотоку в режимі реального часу.

У зв'язку з таким стрімким зростанням потреб підприємств у біометричних системах розпізнавання, аналітики очікують, що зростання інтересу до технологій розпізнавання облич найближчими роками збільшиться ще більше. Основна сфера застосування технології так само буде пов'язана з системами безпеки СКУД та системами моніторингу, але сфера їх використання з кожним роком буде розширюватися.

У зв'язку з цим було визначено **мету магістерської кваліфікаційної роботи** – підвищення ефективності системи контролю доступу співробітників підприємства за допомогою використання штучних нейронних мереж для розпізнавання облич.

Об'єктом дослідження є процеси розпізнавання облич за допомогою нейромережевих систем.

Предметом дослідження є методи, моделі та програмні засоби розпізнавання облич за допомогою штучних нейронних мереж.

Для досягнення поставленої мети потрібно:

- провести аналіз існуючих на ринку видів інтелектуальних систем розпізнавання облич;
- провести аналіз основних видів інформаційних моделей штучних нейронних мереж;
- визначити основні інструментальні засоби для розробки системи;

– спроектувати систему розпізнавання облич співробітників підприємства;
– провести дослідження ефективності системи розпізнавання облич співробітників підприємства.

Апробація результатів дослідження. Основні положення магістерської кваліфікаційної роботи, зокрема інтелектуальна система розпізнавання облич співробітників підприємства, викладені в матеріалах Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія», підсекція «Машинне навчання та штучний інтелект» (2023 р., ЧНУ ім. Петра Могили, м. Миколаїв).

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної сфери

Визначення штучної нейронної мережі. Штучною нейронною мережею можна розуміти як математичну модель, та навіть її програмна реалізація, побудована за принципом нервових клітин живого організму – біологічних нейронних мереж. Саме під час спроби змоделювати процеси, як у мозку людини і виникло це поняття [4].

Система простих процесорів (штучних нейронів), з'єднаних і взаємодіючих між собою є штучною нейронною мережею. Кожен із процесорів мережі має справу з сигналами, які періодично передаються іншим процесорам. За допомогою великої мережі можна вирішувати дуже складні завдання у короткі терміни.

З математичної точки зору, нейронні мережі є способом вирішення нелінійних задач оптимізації. Кібернетика застосовує нейромережеву теорію при рішенні задач адаптивного керування, побудові робото технічних алгоритмів [5].

Програмування нейронної мережі означає навчання самої мережі, а не написання коду програми. Отже, завдяки навчанню мережі мають здатність виявляти залежності між вхідними та вихідними даними, спрощувати результати, узагальнювати, використовувати знання для розкладання складних задач на більш прості.

Біологічний нейрон і штучний нейрон. Людський мозок і його нервова система складаються з нейронів, які з’єднані між собою нервовими волокнами. Електричні імпульси передаються між нейронами за допомогою нервових волокон. Всі дії, які відбуваються з живим організмом, всі пошкодження шкіри, подразнення очей, відчуття болю, процеси мислення є взаємодією між нейронами. Будова біологічного нейрона представлена на рис. 1.1:

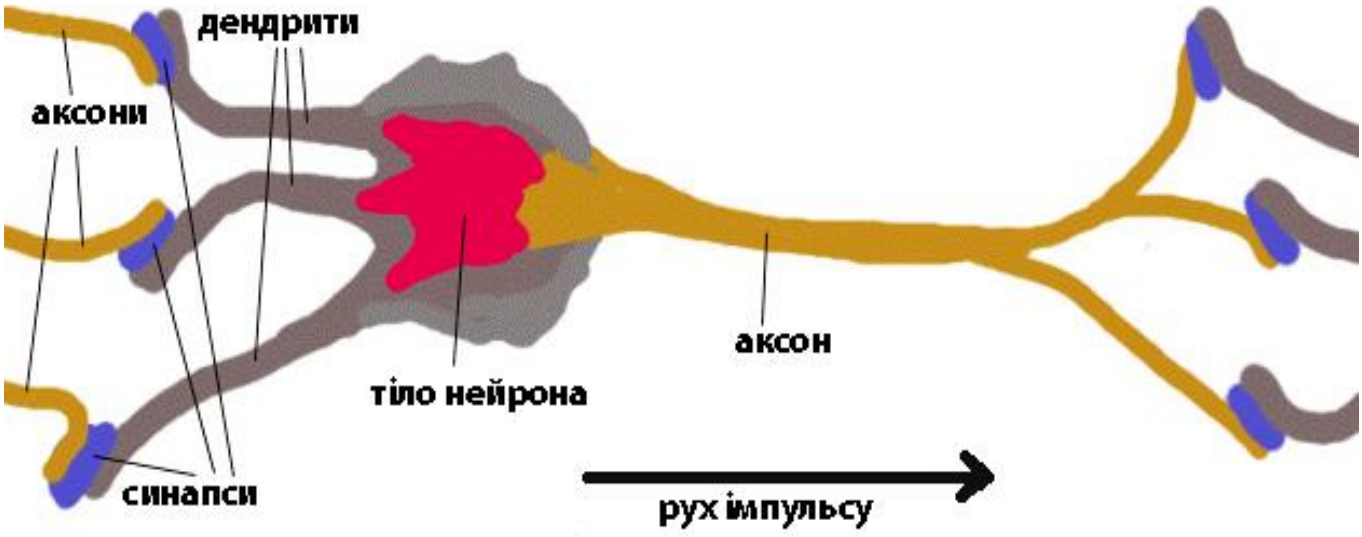


Рис. 1.1 – Біологічний нейрон [6]

Дендрити – приймають імпульси нейрона;

Аксон – передає імпульс нейрона;

Синапси – утворення, які мають вплив на силу імпульсів, для контакту аксону і дендритів.

При проходженні через синапс сила імпульсів піддається змінам в декілька разів (вага синапсу). Коли до нейронів надходять імпульси від декількох дендритів, усі вони сумуються. Якщо сумарний імпульс є більшим ніж мінімальний поріг, нейрон переходить у збуджений стан, формує власний імпульс і посилає його далі по аксону. Поведінка відповідного нейрона змінюється, через те що ваги синапсів можуть змінюватися з часом. Математична модель описаного процесу зображена так (рис. 1.2) [6]:

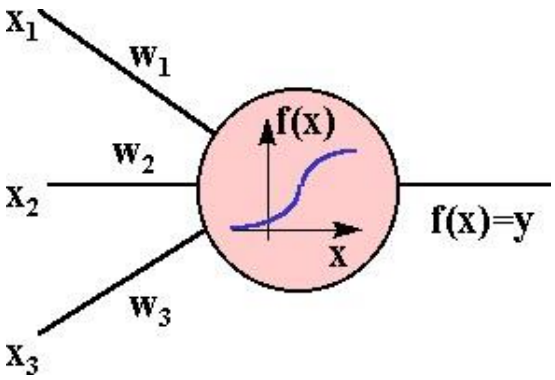


Рисунок 1.2 – Математична модель нейрона

Дана модель описує нейрон з трьома входами (дендритами), де синапси мають ваги w_1, w_2, w_3, до яких поступають сили x_1, x_2, x_3. До нейрона поступають імпульси x_1w_1, x_2w_2, x_3w_3 після проходження синапсів та дендритів.

Одержаний сумарний імпульс

$$x=x_1w_1+x_2w_2+x_3w_3 \quad (1.1)$$

нейрон перетворює у відповідності до передатної функції f(x).

$$y = f(x)=f(x_1w_1+x_2w_2+x_3w_3) \quad (1.2)$$

сила вихідного імпульсу. Підсумовуючи, одержуємо набір чисел xk (вектор), який має вигляд входів. Далі нейрон надає певне число на виході [7].

На вхід штучного нейрону надходить дуже багато сигналів, кожен з яких одночасно є виходом іншого нейрона. Такий вхід помножується на підходящу вагу, потім добутки додаються, визначаючи рівень активації нейрона.

Модель, яка реалізує цю теорію, показана на рис 1.3:

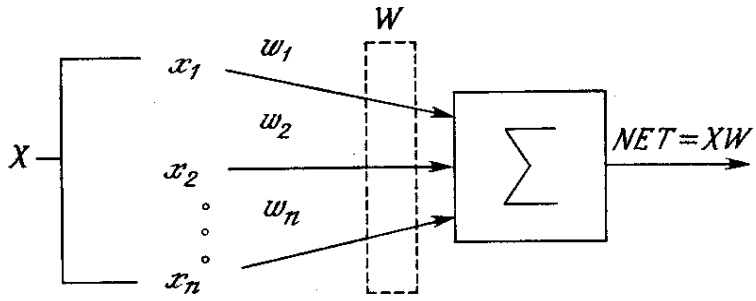


Рисунок 1.3 – Модель активації нейрона

Велика кількість сигналів x_1, x_2,…, x_n, надходить до входу штучного нейрона, потім вони разом позначаються вектором X. Ці сигнали, схожі на ті, які отримує на вхід біологічний нейрон. Далі сигнал також множиться на відповідні ваги w_1, w_2,…, w_n, потім додаються в блоці, який займається підсумуванням ∑. Кожні ваги дорівнюють силі одного синоптичного зв'язку в біологічному нейроні. Вихід, що частіше всього називається NET, створюється з блоку додавання, де виважені елементи складаються математично.

Сигнал *NET* перетворюється лінійною функцією, яка є активаційною. Вона позначається *F* та надає вихідний сигнал *OUT*.

$$OUT=K(NET), \quad (1.3)$$

де *K* – постійна, порогової функції

OUT=1, якщо *NET>T*

OUT = 0 в інших випадках,

T – постійна гранична величина, яка точніше моделює нейронну мережу.

Штучний нейрон з функцією активації показано на рис. 1.4:

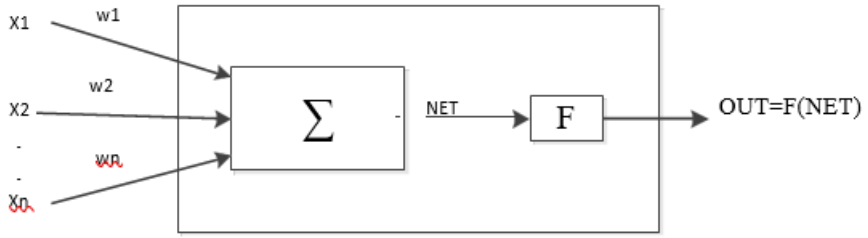


Рисунок 1.4 – Активаційна функція нейрона [8]

Блок, який позначено *F* приймає сигнал *NET* і видає сигнал *OUT*.

F – називається стискаючою функцією, якщо за будь-яких значень *NET* значення *OUT* належать деякому кінцевому інтервалу [9].

Модель штучного нейрона ігнорує велику кількість властивостей біологічного нейрона. Наприклад, часові затримки, що впливають на динаміку системи.

Вхідні сигнали негайно генерують вихідний сигнал. Крім того, штучний нейрон не враховує впливу функції для синхронізації біологічного нейрона.

Проте слід зазначити виняткову схожість між живим нейроном і штучним. Щоб визначити місце нейронних мереж у сфері інформаційних технологій, потрібно звернутися до класифікації систем штучного інтелекту.

Класифікація інтелектуальних систем. Інтелектуальні системи мають деякі особливості:

Розвинені комунікативні навички, які характеризують взаємодію між системою та комп’ютером користувача. Також є можливість звернення до системи з випадковим запитом у діалозі з інтелектуальною системою. При

цьому мова інтелектуальної системи має бути максимально наближеною до природної мови.

Рішення слабо формалізованих задач, тобто задач, що не мають конкретного рішення, але вимагають нестандартного підходу в залежності від ситуації, наявних даних і кінцевого результату. Погано формалізовані задачі ефективно вирішуються з використанням штучних нейронних мереж.

Здатність до отримання знань інтелектуальною системою з досвіду який був накопичений у конкретних ситуацій – самонавчання . Попереднє навчання системи вимагає оброблених вихідних даних. За наведеними характеристиками інтелектуальні системи можна поділити на наступні (табл. 1.1):

Таблиця 1.1 – Види інтелектуальних систем

Вид інтелектуальної системи	Тип інтелектуальної системи
Системи з комутативними здібностями	інтелектуальні бази даних; природно-мовні інтерфейси; гіпертекстові системи; контекстні довідкові системи; когнітивна графіка.
Експертні системи	класифікуючі системи; довизначальні системи; трансформуючі системи; багатоантенні системи.
Самонавчальні системи	індуктивні системи; нейронні сіті; Системи на прецедентах; Інформаційні сховища.
Адаптивні системи	CASE – технології; Компонентна технологія.

Інтелектуальні бази даних відрізняються від звичайних баз даних тим , що в них є можливість вибору необхідної інформації за запитом, яка може не зберігатися явно, але може бути отримана з бази даних.

Звичайно, мовний інтерфейс перетворює конструкції природної мови на внутрішнє представлення знань на машинному рівні. Використовується для доступу до інтелектуальних баз даних, контекстного пошуку текстової інформації у документах, голосових команд в системах управління, машинного перекладу з іноземних мов.

Гіпертекстові системи використовуються в базах текстової інформації, де потрібен пошук за ключовими словами, і мають більш складну семантичну організацію ключових слів.

У контекстних довідкових системах користувач описує проблему (ситуацію), а система уточнює її за допомогою додаткових діалогів та шукає відповідні для цієї ситуації рекомендації. Такі системи створюються як доповнення до систем документації і належать до класу систем розширення інформації.

Для управління операційними процесами використовуються системи когнітивної графіки. Графічні зображення в наочно-цілісному вигляді описують багато параметрів досліджуваної ситуації.

Експертні системи призначені для рішення задач на базі накопичених знань, які відображає досвід експертів у проблемній області.

Мультиагентні системи — це динамічні системи, які характеризуються інтеграцією в базі знань декількох джерел знань різного роду, які обмінюються між собою отриманими результатами на динамічній основі.

Системи самонавчання базуються на методах автоматичної класифікації прикладів ситуацій реальної практики.

Характерними рисами самонавчальних систем є:

– системи самонавчання «з учителем», коли для кожного прикладу в явному вигляді задається значення ознаки його належності до певного класу ситуацій (класоутворюючої ознаки);
– системи самонавчання «без вчителя», коли система сама виділяє класи ситуацій за ступенем близькості значень класифікаційних ознак.

Індуктивні системи узагальнюють приклади за принципом від окремого до загального, а процес узагальнення здійснюється таким чином:

– класифікаційна ознака вибирається з набору заданих (послідовно або за правилом);
– за значенням виділеної ознаки множина прикладів розбивається на підмножини;
– виконується перевірка належності прикладу до одного класу;
– якщо будь-яка підмножина прикладів належить до одного підкласу, тобто всі приклади підмножини мають однакове значення класоутворюючої ознаки, то процес класифікації завершується (інші класифікаційні ознаки не розглядаються);
– для підмножин прикладів з невідповідним значенням класоутворюючої ознаки процес класифікації продовжується, починаючи з пункту 1.

Нейронні мережі – це паралельні обчислювальні інструменти, які складаються з багатьох простих процесорів, які періодично отримують і посилають сигнали іншим процесорам.

Якщо, звернутися до інтелектуальних інформаційних систем з точки зору задач, що вирішуються. Потрібно виділити наступні системи:

– системи управління та довідкові системи;
– системи комп'ютерної лінгвістики;
– системи розпізнавання;
– ігрові системи;
– системи створення інтелектуальних інформаційних систем [10].

На рис. 1.5 представлена класифікація інтелектуальних інформаційних систем на основі вирішуваних завдань:

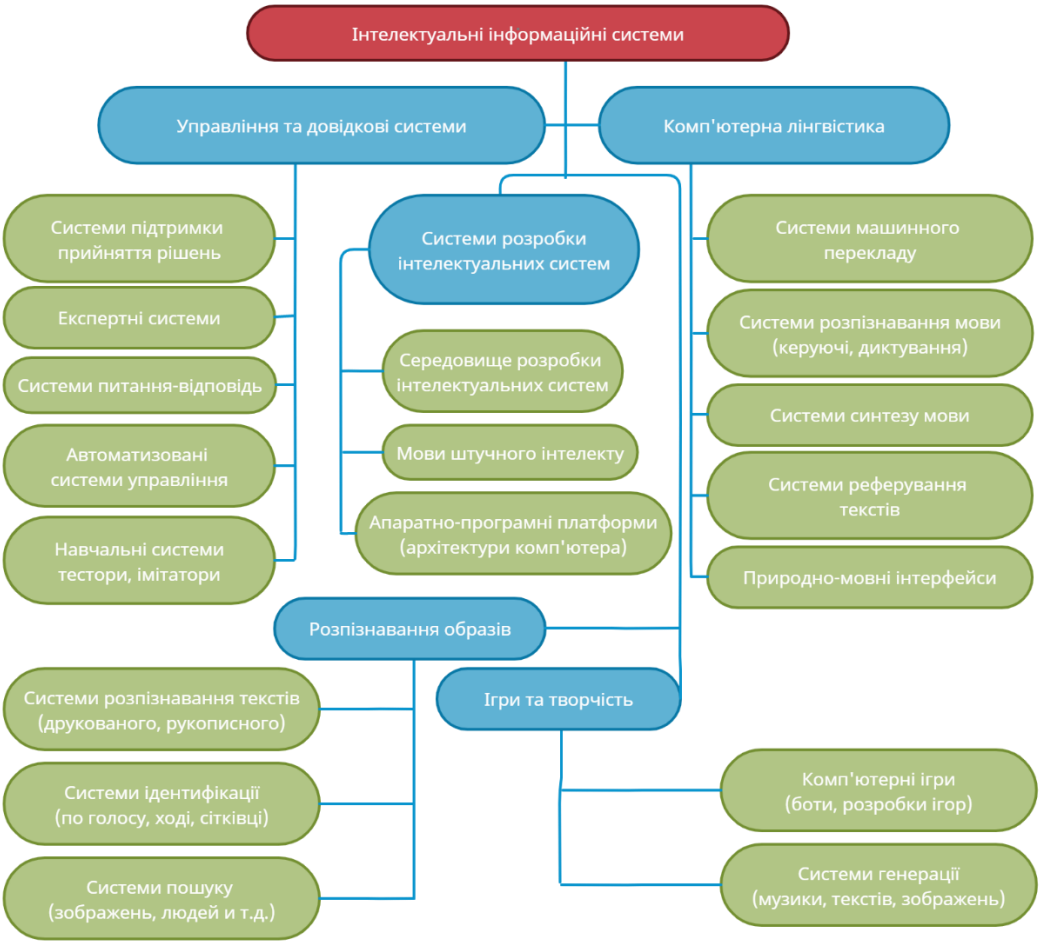


Рисунок 1.5 – Класифікація ІС по розв'язуваним завданням [11]

Системи можуть вирішувати не одне, а кілька завдань, або вирішувати ряд інших задач у процесі вирішення однієї задачі.

Інтелектуальні системи також класифікують за таким критерієм, як використовувані методи. Тут потрібно зазначити м'який, жорсткий і гібридний методи (див. рис. 1.6):

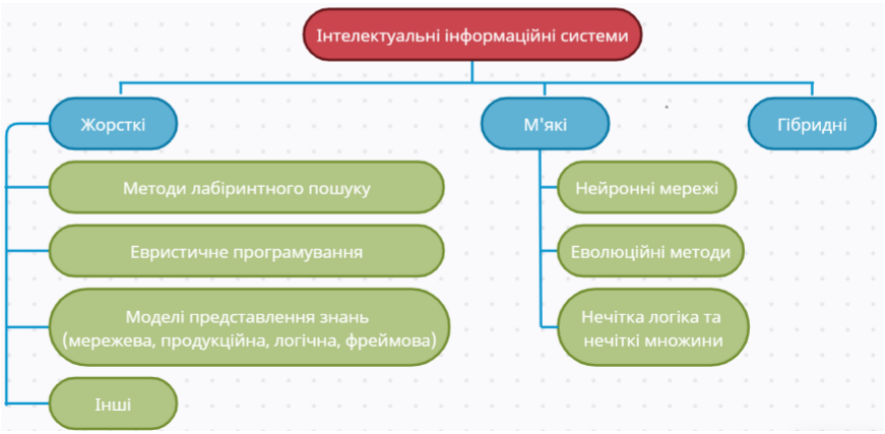


Рисунок 1.6 – Класифікація ІС методами

М'які розрахунки (Soft Computing) – це комплексна комп’ютерна методологія, яка використовує нечітку логіку, генетичні обчисленнях, нейрокомп’ютинг та імовірнісні обчислення.

Жорсткі розрахунки – це стандартні комп'ютерні розрахунки (які не належать до м'яких розрахунків).

Гібридні системи – це системи, які використовують більше однієї комп’ютерної технології (у випадку інтелектуальних систем – технології штучного інтелекту) [12].

Класифікація інтелектуальних систем охоплює дуже велику теоретичну базу знань у різних галузях науки. Створення та впровадження інтелектуальних інформаційних систем – це складний процес від початкового до завершального етапу.

Системи розпізнавання облич. Процесом розпізнавання облич зазвичай називають комплекс різноманітних задач, які служать для ідентифікації людини за цифровим зображенням або фрагментом відео. У загальному вигляді цей процес виглядає так: після отримання системою зображення з камери за допомогою алгоритмів визначаються межі обличчя (етап виявлення). Далі йде етап розпізнавання, на якому обличчя трансформується (змінюється його яскравість, вирівнюється, масштабується тощо) і надається певний заданий вигляд. Після цього функції обчислюються та безпосередньо порівнюються з контрольними показниками, вбудованими в базу даних. Цей останній етап порівняння називається ідентифікацією або перевіркою залежно від системи.

Перевірка: порівняння зразків за схемою «1:1». Для ідентифікації особи системі потрібно порівняти біометричний зразок з одним біометричним шаблоном, який знаходиться в базі даних, і відповідає на запитання «Ця особа та, з шаблоном якої її порівнювали?».

Ідентифікація: порівняння зразків за схемою «1:N». Для визначення особи система порівнює біометричний зразок з усіма шаблонами обличчя, які є

знаходяться в базі, і відповідає на запитання «хто це?». На рисунку 1 зображено загальний алгоритм розпізнавання облич із зображення.

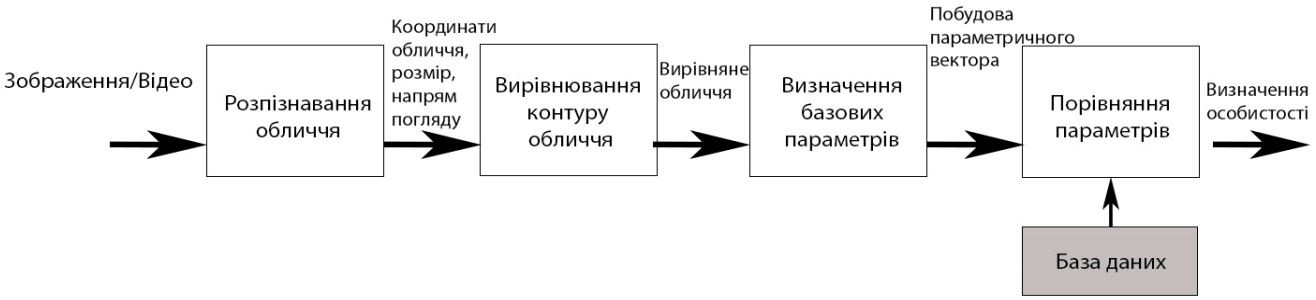


Рисунок 1.7 – Алгоритм розпізнавання облич із зображення

1.2 Огляд та аналіз наявних аналогів та публікацій

З кожним роком зростання інтересу до технологій розпізнавання облич помітно зростає. На сьогоднішній день вже існує кілька десятків систем розпізнавання, які застосовують у різних сферах людського життя. Лідерами серед компаній, що розробляють такі системи, вважаються:

корпорація NEC (Японія), Cognitec Systems GMBH (Німеччина), Neurotechnology (Литва).

Система «FaceVACS» компанії «Cognitec Systems». «FaceVACS-VideoScan» – просте у використанні програмне забезпечення розпізнавання облич по відеопотоку в реальному часі, яке пропонує компанія «Cognitec Systems» [13].

Система "FaceVACS-VideoScan" складається з декількох системних компонентів: відеосервера керуючого відеопотоками; сервера відеосканування, що координує всі компоненти системи і виконує основні біометричні операції; обчислювального вузла, що використовується для розподілу обчислювального навантаження; інтерфейсу користувача; диспетчера сигналів одержувача повідомлення про події та обслуговуючого мобільні пристрої; операційну базу даних; та комплекту інтеграторів.

На сьогоднішній день технологія FaceVACS використовує алгоритм розпізнавання облич B10T9. Цей алгоритм, стійкий до змін міміки, поворотів

обличчя (на ±15º) часткового його закриття, використання окулярів та зміни освітлення.

Крім того, система FaceVACS має такі особливості:

– можливість одночасного відстеження кількох облич;

– порівняння облич відбувається у реальному часі;

– можливість відображення та відправлення статистики про потоки;

– підтримка інтерактивної реєстрації з нерухомого;

– застосування C++ API та Web Services API.

Система NEC's Face Recognition компанії NEC. NEC's Face Recognition – одна з передових систем розпізнавання облич, розроблена японською компанією NEC, що дозволяє ідентифікувати людей по кадрах багаторічної давності і навіть, якщо людина перебуває в окулярах або гримасує. Всі розпізнані особи зберігаються у базі даних, тому в разі потреби можна підняти всю історію відеореєстрації та переглянути дату та час будь-якого збереженого зображення [14].

Технологія NEC перевершує безліч інших систем розпізнавання своєю точністю та швидкістю. Вона має хороші показники продуктивності в різних ситуаціях, у тому числі під час роботи з відео низької якості та сильно стислими зображеннями. NEC аналізує індивідуальні особливості особи (розмір, форму зіниць, лінії носа та рота), їх взаємне розташування, і знаходить потім за цією інформацією відповідну людину в базі даних.

Система включає кілька модулів, що реалізують такі алгоритми:

Використовується метод узагальненої відповідності (GMFD), який забезпечує високу швидкість детектування та високу точність розпізнавання обличчя. Метод GMFD заснований на нейронних мережах та здійснює попередній пошук пар очей;

Алгоритм PSM (Perturbation Space Method), дозволяє ефективно справлятися з труднощами пов'язаними з розташуванням обличчя в кадрі (обличчя під нахилом або деяким кутом).

Метод ARBM (Adaptive Regional Blend Matching), який зменшує вплив невеликих змін на обличчі (наприклад, зміни виразу обличчя, наявність окулярів, головного убору) на точність розпізнавання.

Система розпізнавання облич NeoFace має такі особливості:

– можливість спостереження та контролю в реальному часі;

– ідентифікація на основі індивідуальних рис особи;

– множинне розпізнавання;

– можливість пошуку подій по базі даних;

– ведення журналу зображень облич;

– стійкість до повороту особи на ±15º та нахилу голови до 45º у будь-якому. напрямку від фронтального положення;

– "Drag and Drop" управління;

– масштабований та необмежений розмір бази даних;

– незалежне розпізнавання напряму погляду та характеристик обличчя. (окуляри, борода та вираз обличчя).

*Система "LUNA SDK" компанії "VisionLabs"***.** «LUNA SDK» – спеціалізоване ПЗ, супроводи та розпізнавання облич людей на цифрових фотографіях або відеопотоці від компанії "VisionLabs". Двигун має одні з найкращих у світі показників повноти та точності розпізнавання в реальних умовах [15].

Процес розпізнавання облич у LUNA SDK працює на основі глибинних нейронних мереж та складається з кількох ключових етапів: визначення положення та розмірів усіх облич (детекції); визначення розташування характерних рис особи та трансформації їх у стандартизовану форму (вирівнювання); вилучення дескрипторів (числових векторів, що підсумовують характерні ознаки особи); порівняння облич з базою зображень та запобігання підміні облич.

Технічні характеристики та особливості LUNA SDK:

– повністю розроблена на c++;

– можливість спостереження та контролю в реальному часі;

– можливість працювати в режимі багатопоточності;

– використання алгоритму розпізнавання облич на основі глибинних нейронних мереж;

– оптимізоване керування пам'яттю;

– компактні дескриптори, які забезпечують обчислювальну ефективність;

– підтримка різних платформ, включаючи windows та linux.

Дескриптори LUNA навчені на мільйонах облич з різних джерел і забезпечують високу точність у різних умовах, наприклад, у банках, системах відеоспостереження та соціальних мережах. Нижче наведено оцінку точності системи LUNA на реальному прикладі з клієнтської бази даних.

Система «VeriLook SDK» компании «Neurotechnology». "VeriLook SDK" - технологія ідентифікації облич, розроблена компанією "Neurotechnology". Є системою виявлення облич з можливістю одночасного множинного розпізнавання людей присутніх у кадрі та швидкої ідентифікації облич (знаходить до 100000 облич за секунду). VeriLook SDK доступна у вигляді комплекту для розробки програмного забезпечення та підтримує широкий вибір пристроїв на Windows Linux, Mac OS X, iOS та Android [16].

Алгоритм VeriLook реалізує локалізацію обличчя з використанням алгоритмів обробки цифрових зображень, заснованих на глибоких нейронних мережах.

Основними перевагами системи VeriLook є відсутність необхідності контакту із засобами сканування та швидке впровадження функцій біометричної ідентифікації у прикладні системи замовника, а також існує низка інших переваг:

– одночасна обробка кількох облич;

– гендерна класифікація. за бажанням, підлога може бути визначена для кожної людини на зображенні;

– живе розпізнавання обличчя. verilook визначає, чи є особа у відеопотоці «живою» або фотографією;

– розпізнавання емоцій. verilook аналізує шість основних емоцій: гнів, огиду, страх, щастя, смуток та подив;

– атрибути особи. verilook може бути налаштовано для виявлення певних атрибутів під час отримання обличчя - посмішки, відкритого рота, закритих очей, окулярів, бороди або вусів;

–визначення якості зображення обличчя. поріг якості може використовуватися під час реєстрації особи, щоб гарантувати, що у базі даних зберігатимуться лише шаблони кращої якості.

– кілька зразків однієї й тієї ж особи. ці зразки можуть бути зараховані з різних джерел та в різний час, що дозволяє покращити якість зіставлення;

– ідентифікаційна здатність. функції verilook можуть використовуватися в порівнянні 1-к-1 (перевірка), а також в режимі 1-многим (ідентифікація);

– малий шаблон обличчя. шаблон особи може бути розміром від 4 кг;

– особливості режиму узагальнення. цей режим генерує колекцію узагальнених функцій особи з кількох зображень того самого об'єкта;

– алгоритм verilook здатний зіставляти межі, захоплені в інфрачервоному спектрі.

1.3 Постановка задачі

Метою магістерської кваліфікаційної роботи є синтез та дослідження нейромережевої системи для розпізнавання облич співробітників підприємства.

Реалізація поставленої мети передбачає вирішення наступних завдань:

− вибір інструментальних засобів розробки нейромережевої системи;

− моделювання та проектування системи для розпізнавання облич.

− розробка нейромережевої системи для розпізнавання облич співробітників підприємства.

Висновки до розділу 1

В даному розділі було розглянуте поняття штучних нейронних мереж та процес розпізнавання облич. Крім того детально проаналізовано наявні системи розпізнавання облич, які є лідерами у світі.

З кожним роком зростання інтересу до технологій розпізнавання облич помітно зростає. На сьогоднішній день вже існує кілька десятків систем розпізнавання, які застосовують у різних сферах людського життя.

Для досягнення поставленої мети в даній магістерській кваліфікаційній роботі потрібно розв’язати наступні задачі:

- провести аналіз існуючих на ринку видів інтелектуальних систем розпізнавання облич;
- провести аналіз основних видів інформаційних моделей штучних нейронних мереж;
- визначити основні інструментальні засоби для розробки системи;
- спроектувати систему розпізнавання облич співробітників підприємства;
- провести дослідження ефективності системи розпізнавання облич співробітників підприємства.

2 МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ОБЛИЧ

2.1 Вибір інструментальних засобів

Для розробки системи розпізнавання облич було обрано інтерпретовану об'єктно-орієнтовану мову програмування високого рівня Python v. 3.6.

Python – це мова програмування, яка широко використовується в інтернет-додатках, розробці програмного забезпечення, науці даних і машинному навчанні (ML). Розробники використовують Python, тому що він ефективний, простий у вивченні та працює на різних платформах. Програми мовою Python можна завантажити безкоштовно, вони сумісні з усіма типами систем та підвищують швидкість розробки [17].

Мова Python має такі переваги:

- розробники можуть легко читати та розуміти програми на Python, оскільки мова має базовий синтаксис, схожий на синтаксис англійської;
- Python допомагає розробникам бути більш продуктивними, оскільки вони можуть писати програми на Python, використовуючи менше рядків коду, ніж іншими мовами;
- Python має велику стандартну бібліотеку, що містить багаторазові коди практично для будь-якого завдання. В результаті розробникам не потрібно писати код із нуля;
- розробники можуть легко поєднувати Python з іншими популярними мовами програмування: Java, C та C++;
- активна спільнота Python складається з мільйонів розробників з усього світу. У разі виникнення проблем співтовариство допоможе в їх вирішенні;
- крім того, в Інтернеті є безліч корисних ресурсів для вивчення Python. Наприклад, ви можете легко знайти відеоролики, навчальні посібники, документацію та посібники для розробників;

– Python можна переносити на різні операційні системи: Windows, MacOS, Linux і Unix;

Мова Python має кілька стандартних прикладів використання при розробці додатків, серед яких:

Веб-розробка на стороні сервера. Веб-розробка на стороні сервера включає складні серверні функції, за допомогою яких веб-сайти відображають інформацію для користувача. Наприклад, веб-сайти повинні взаємодіяти з базами даних та іншими веб-сайтами, а також захищати дані під час їх надсилання по мережі.

Python корисний при написанні серверного коду, оскільки він пропонує безліч бібліотек, що складаються із попередньо написаного коду для складних серверних функцій. Також розробники використовують широкий спектр платформ Python, які надають всі необхідні інструменти для більш швидкого та простого створення інтернет-додатків. Наприклад, розробники можуть створити «скелет» інтернет-програми за лічені секунди, тому що їм не потрібно писати код з нуля. Потім можна протестувати за допомогою інструментів тестування платформи незалежно від зовнішніх інструментів тестування.

Автоматизація за допомогою скриптів Python. Мова скриптів – це мова програмування, яка автоматизує завдання, які зазвичай виконують люди. Програмісти широко використовують скрипти Python для автоматизації багатьох повсякденних завдань, серед яких:

– одночасне перейменування великої кількості файлів
– перетворення файлу на інший тип файлу
– видалення повторюваних слів у текстовому файлі
– виконання базових математичних операцій
– надсилання повідомлень електронної пошти
– завантаження контенту
– виконання базового аналізу журналів
– пошук помилок у кількох файлах

Наука про дані та машинне навчання. Наука про дані отримує цінну інформацію з даних, а машинне навчання (ML) дозволяє комп'ютерам автоматично вчитися на даних і робити точні прогнози. Фахівці з роботи з даними використовують Python для вирішення наступних завдань:

- виправлення та видалення неправильних даних (очищення даних);
- вилучення та вибір різних характеристик даних;
- розмітка даних додає даним значні імена;
- пошук статистичної інформації у даних;
- візуалізація даних за допомогою діаграм та графіків: лінійних діаграм, стовпчастих діаграм, гістограм та кругових діаграм.

Фахівці з роботи з даними використовують бібліотеки Python ML для моделей машинного навчання та створення класифікаторів, які точно класифікують дані. Класифікатори на основі Python використовуються в різних областях і застосовуються для виконання таких завдань, як класифікація зображень, тексту та мережевого трафіку, розпізнавання мови та розпізнавання осіб. Фахівці роботи з даними також використовують Python для глибокого навчання — передової техніки машинного навчання.

Розробка програмного забезпечення. Розробники програмного забезпечення часто використовують Python для різних завдань розробки та програмних додатків, серед яких:

- відстеження помилок у програмному коді;
- автоматичне складання програмного забезпечення;
- управління програмними проектами;
- розробка прототипів програмного забезпечення;
- розробка настільних додатків з використанням бібліотек графічного інтерфейсу користувача (дпі);
- розробка ігор: від простих текстових ігор до складних ігор.

Мова Python унікальна завдяки наступним особливостям:

Python є мовою, що інтерпретується, тобто він виконує код рядково. Якщо в коді програми є помилки, вона перестає працювати. Це дозволяє програмістам швидко знайти помилки у коді. Python використовує слова, подібні до слів англійської мови. На відміну від інших мов програмування, у Python не використовуються фігурні дужки. Замість них застосовується відступ. Програмістам не потрібно оголошувати типи змінних під час написання коду, тому що Python визначає їх під час виконання. Ця функція дозволяє писати програми на Python значно швидше. Python ближче до природних мов, ніж низка інших мов програмування. Завдяки цьому програмістам не потрібно турбуватися про його базову функціональність, наприклад про архітектуру та управління пам'яттю. Python розглядає всі елементи як об'єкти, але також підтримує інші типи програмування (наприклад, структурне та функціональне програмування).

Вибір Python IDE. Інтегроване середовище розробки (IDE) – це програмне забезпечення, яке надає розробникам інструменти для написання, редагування, тестування та налагодження коду.

Найпопулярніші Python IDE

PyCharm – результат праці JetBrains, чеської компанії з розробки програмних інструментів. У програми є як безкоштовна версія для невеликих додатків, так і платна професійна версія, що підходить для створення великих додатків Python [18].

Інтегроване середовище розробки та навчання (IDLE) – це інтегроване середовище розробки Python, встановлене за умовчанням. Середовище розроблене лише на Python з використанням набору інструментів Tkinter GUI.

Spyder – це IDE з відкритим вихідним кодом, яку використовують багато фахівців та аналітики даних. Вона застосовується для всебічної розробки з використанням функцій розширеного аналізу даних, візуалізації та налагодження [19].

Atom – це безкоштовний редактор, розроблений GitHub, який підтримує кодування багатьма мовами програмування, зокрема Python. Atom дозволяє працювати з GitHub – веб-сайтом, на якому можна централізовано зберігати свій код [20].

Для розробки системи було обрано Python IDE під назвою PyCharm, через такі переваги:

- автоматичне завершення та перевірка коду;
- обробка та швидке усунення помилок;
- чистка коду без зміни функціональних можливостей;
- підтримка платформ інтернет-додатків, таких як django та flask;
- підтримка інших мов програмування, таких як javascript, coffeescript, typescript, angularjs та node;
- наукові інструменти та бібліотеки, такі як matplotlib та numpy;
- можливість запуску, налагодження, тестування та розгортання програм на віддалених віртуальних машинах;
- відладчик для пошуку помилок у коді, профільувальник для виявлення проблем з продуктивністю та засіб запуску модульних тестів;
- підтримка баз даних.

Графічний інтерфейс. Графічний інтерфейс було створено за допомогою PyQt5 – набору розширень графічного фреймворку Qt для мови програмування Python, виконаний як розширення Python [21].

PyQt розроблено британською компанією Riverbank Computing. PyQt працює на всіх платформах, що підтримуються Qt: Linux та інші UNIX-подібні ОС, macOS та Windows. Існує 3 версії: PyQt6, PyQt5 та PyQt4, що підтримують відповідні версії Qt. PyQt поширюється під ліцензіями GPL (2 та 3 версії) та комерційною.

Для розробки системи було обрано саме версію PyQt5. Вибір саме цієї версії обумовлений тим, що PyQt4 вже застаріла версія, а PyQt6 нова та не має значної кількості прикладів роботи з нею. В той же час PyQt5 – стабільна робоча версія, для

якої у вільному доступі є велика кількість прикладів роботи з нею та її застосування.

PyQt практично повністю реалізує можливості Qt. Це понад 600 класів, понад 6000 функцій та методів, включаючи:

- набір віджетів графічного інтерфейсу;
- стилі віджетів;
- доступ до баз даних за допомогою SQL (ODBC, MySQL, PostgreSQL, Oracle);
- QScintilla, заснований на Scintilla віджет текстового редактора;
- підтримку інтернаціоналізації (i18n);
- парсер XML;
- підтримку SVG;
- інтеграцію з WebKit, двигуном рендерингу HTML;
- підтримку відтворення відео та аудіо.

Qt – фреймворк для розробки кроссплатформенного програмного забезпечення мовою програмування C++. Для багатьох мов програмування існують бібліотеки, що дозволяють використовувати переваги Qt: Python - PyQt, PySide, Ruby - QtRuby, Java - QtJambi, PHP - PHP-Qt та інші [22].

За допомогою Qt можна запускати написані програми у ньому майже на всіх операційних систем шляхом простої компіляції програми для кожної системи без зміни вихідного коду. Включає всі базові класи, що можуть знадобитися для розробки прикладного програмного забезпечення, від елементів графічного інтерфейсу користувача до класів мереж, баз даних і XML. Він повністю об'єктно-орієнтований, розширюваний і підтримує методи компонентного програмування.

PyQt працює з Qt Designer (Qt Creator) – дизайнер графічного інтерфейсу користувача. За допомогою програми pyuic.exe генерується код Python із .ui файлів, створених у Qt Designer. Через це PyQt є дуже корисним інструментом для створення графічного інтерфейсу програми. Крім того, є можливість додати нові графічні елементи керування, написані на Python, Qt Designer.

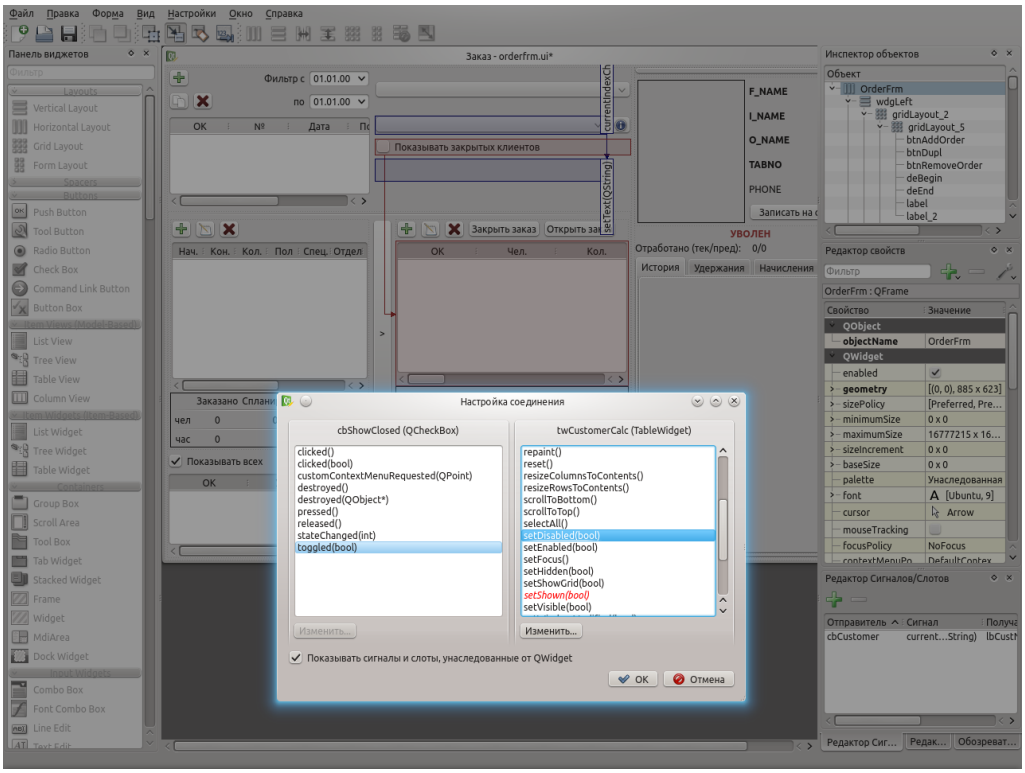


Рисунок 2.1 – Інтерфейс Qt Designer

За допомогою великої кількості інструментів у Qt Designer можна створювати графічні інтерфейси. Є панель інструментів «Панель віджетів», в якій доступні для використання елементи інтерфейсу – віджети, такі як, наприклад, Label, Widget, PushButton і купа інших. Кожний віджет має власний набір властивостей, який визначається класом бібліотеки Qt. Властивість віджета можна змінити використавши «Редактор властивостей». Кожен клас потужності віджетів має власний спеціалізований редактор. Вагливою особливістю Qt Designer є робота з візуальним редагуванням сигналів і слотів. Так, наприклад, можна підключити сигнал, який генерується від перемикання стану віджета CheckBox, до слота, що відповідає наявності іншого віджета [23].

2.2 Розпізнавання зображень

OpenCV – opensource-бібліотека для комп'ютерного зору, обробки зображень та чисельних алгоритмів із досить широким функціоналом. Спочатку вона була написаний мовою C/C++, а тепер є підтримка Java, Matlab, Ruby, Lua, Python та інших мов. Вона має підтримані машинним навчанням методи розширеного виявлення облич. Алгоритми виявляють обличчя на зображенні, розбиваючи його

на тисячі шаблонів і характеристик, яким воно відповідає. Завдання на відповідність таких ознак називаються класифікаторами. В цілому, OpenCV – невід'ємна частина стека щодо розробки штучного інтелекту та систем комп'ютерного зору [24].

Можливості OpenCV

Кадрування. Одна з основних можливостей, яку всі дуже часто використовують. OpenCV – доволі простий і зрозумілий API: якщо потрібно обрізати частину вихідного малюнку – треба вказати координати точок для кадрування, кадруємо і продовжуємо роботу.

Рисунок 2.2 – Приклад кадрування зображення

Робота із розмірністю. Є можливість зменшувати рисунки до потрібних розмірів. Це використовується у створенні датасетів для навчання згорткових нейронних мереж, щоб зменшити навантаження на систему, та для зручності в роботі.

Рисунок 2.3 – Приклад зміни розміру

Повертання зображення. Є можливість перевертати вихідний рисунок у будь-якому положенні, віддзеркалювати тощо.

Рисунок 2.4 – Приклад повертання зображення

Також можна перевести кольорове зображення у чорно-біле/градація сірого. Для цього існують декілька відомих алгоритмів, коли є потрібність у виділенні

деяких важливих деталей на рисунку тонко налаштовуючи поріг спрацьовування. Такий підхід у роботі використовується часто.

Рисунок 2.5 – Переведення кольорового зображення у чорно-біле

Розмиття та згладжування. Така опція потрібна, для того щоб мати можливість заблюрити чітке зображення, наприклад обличчя людини, номер автівки, тощо. Але навпаки ця функція не працює – для цього потрібні інші алгоритми.

Рисунок 2.6 – Розмиття/згладжування зображення

Малювання прямокутників та ліній. Прямокутники потрібні для того щоб визначати межі об'єктів під час розмітки датасетів для систем розпізнавання, та підписуємо Label, щоб показати, які об'єкти змогла класифікувати система розпізнавання. Також можна малювати й різні інші геометричні фігури – не тільки прямокутника – а також зафарбовувати їх, змінювати товщину та колір ліній.

Рисунок 2.7 – Приклад малювання прямокутників та ліній

Написання текстів на малюнках. Ця функція використовується при роботі з відеофайлами, коли потрібно динамічно показати якісь параметри або те, що відбувається на екрані в даний момент: частоту кадрів відео потоку, кількість відстежуваний об'єктів в реальному часі в системах підрахунку, додаткова інформація про налагодження.

Рисунок 2.8 – Додавання тексту на зображення

Розпізнавання облич. OpenCV має доволі добрий функціонал для виявлення (ідентифікації) об'єктів на зображенні. Це реалізовано з допомогою раніше навченим моделям на основі нейронних мереж. У наведеному нижче прикладі навчена модель використовувалася для визначення фронтального зображення на фотографії (допускається поворот голови приблизно до 30 градусів). З OpenCV можна дуже швидко розпізнавати на зображенні обличчя. При цьому ідентифікація особи не виконується, але можна з припустити, що людина знаходиться в цій частині кадру.

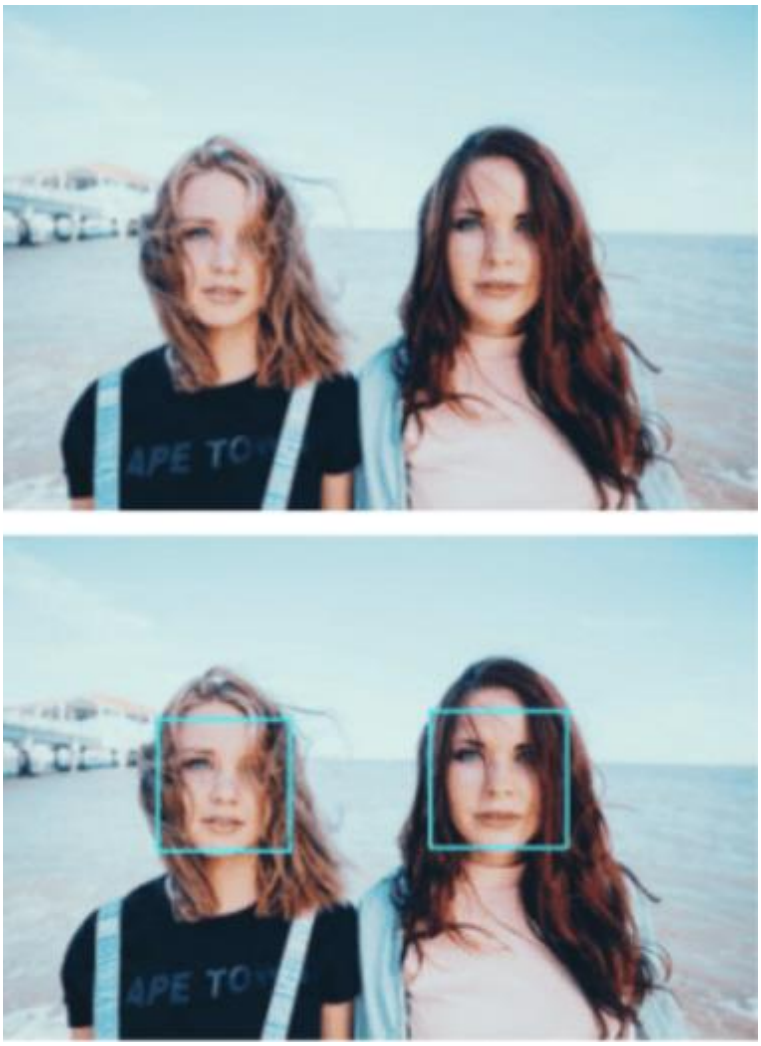


Рисунок 2.9 – Розпізнавання облич

Для пошуку облич використовується принцип «ковзного вікна» (метод Віоли-Джонса). Для кожної області зображення, через яку проходить вікно, обчислюється власне значення Хаара. Наявність або відсутність об’єкта у вікні визначається різницею між значенням ознаки та вивченим порогом. Вікно ніби ковзає по всьому зображенню. Після кожного проходу зображення воно масштабується, щоб знайти обличчя більшого масштабу.

Ознака Хаара складається з суміжних прямокутних областей. Вони розташовуються на зображенні, потім інтенсивності пікселів у регіонах підсумовуються, а після цього обчислюється різниця між сумами. Ця різниця буде значенням для деякої ознаки, деяким чином розташованої на зображенні.

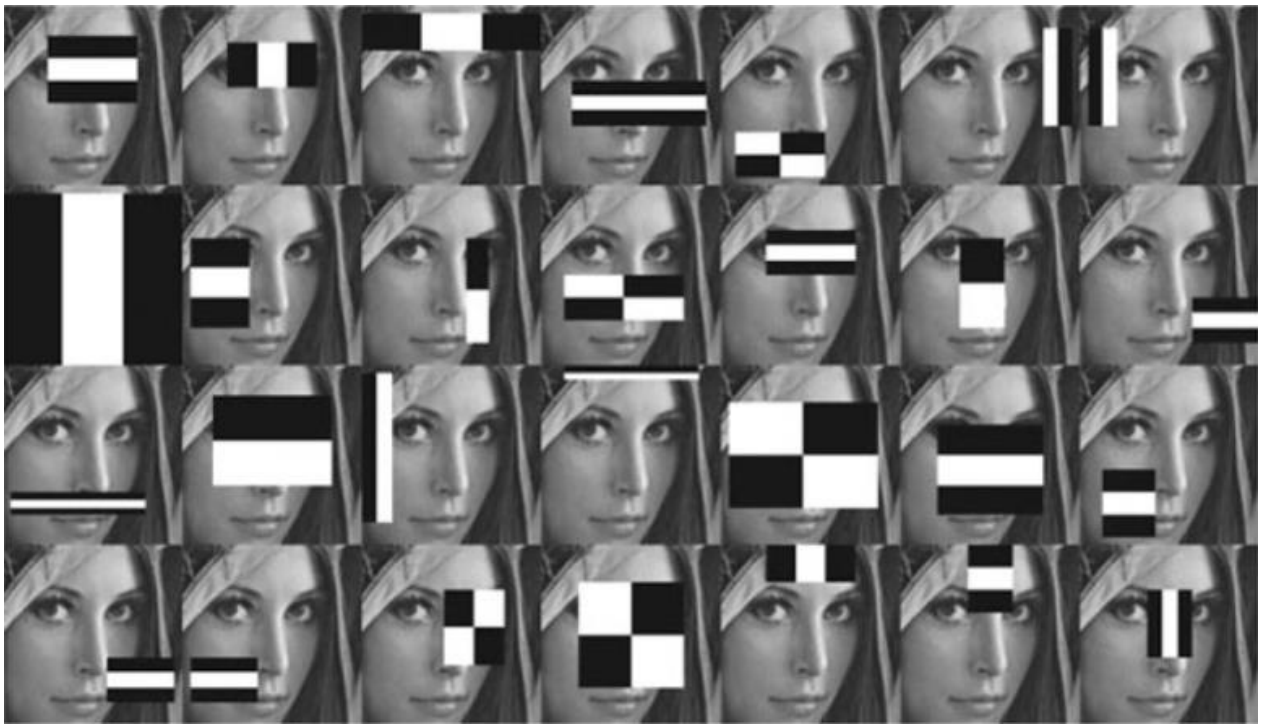


Рисунок 2.10 – Ознака Хаара

Те, що область в районі очей темніша, ніж область в районі щік є спільним для всіх зображень облич. Отже, загальною ознакою Хаара для облич є два суміжні прямокутні регіони, що лежать на щоках і очах [23].

Рисунок 2.11 – Суміжні прямокутні регіони

Саме розпізнавання обличчя OpenCV працює дуже швидкісно – долі секунди. Ця модель доволі складна та довга в навчанні, але працює вона більш швидко, ніж нейромережі які використовують TensorFlow. Також для якісної роботи не вимагають графічний процесор.

Face Recognition Library – ще одна opensource–бібліотека побудована на бібліотеці dlib написаній на C++. Серед особливостей висока точність, дуже корисна у використанні, її легко поставити на сервер не використовуючи потужний GPU. Face Recogn має досить прості вимоги для роботи.

Пошук обличчя на фото. На вході подається зображення і шукаємо обличчя майже, як і в OpenCV, єдине, що в цій бібліотеці дозволяється поворот голови більше ніж на 30 градусів.

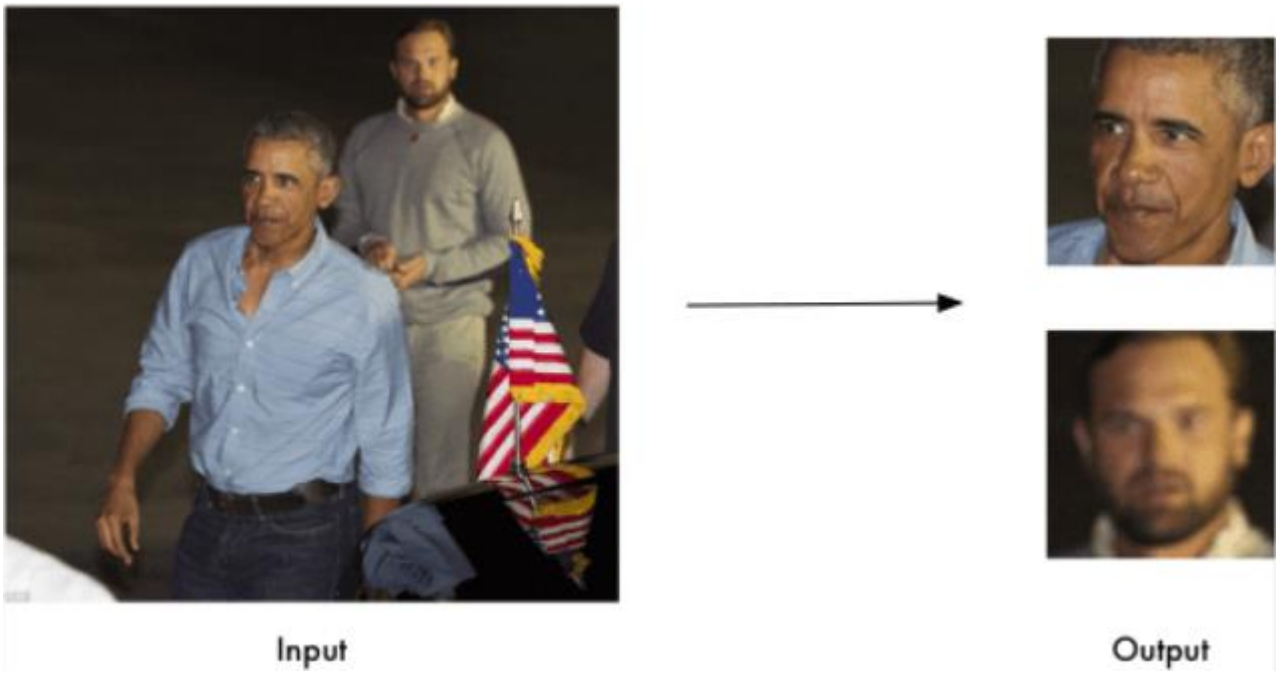


Рисунок 2.12 – Пошук обличчя на фото

Пошук рис обличчя на фотографіях. Розпізнавання фотографій із важливими для ідентифікації контурами та рисами обличчя (очі, ніс, рот, підборіддя).

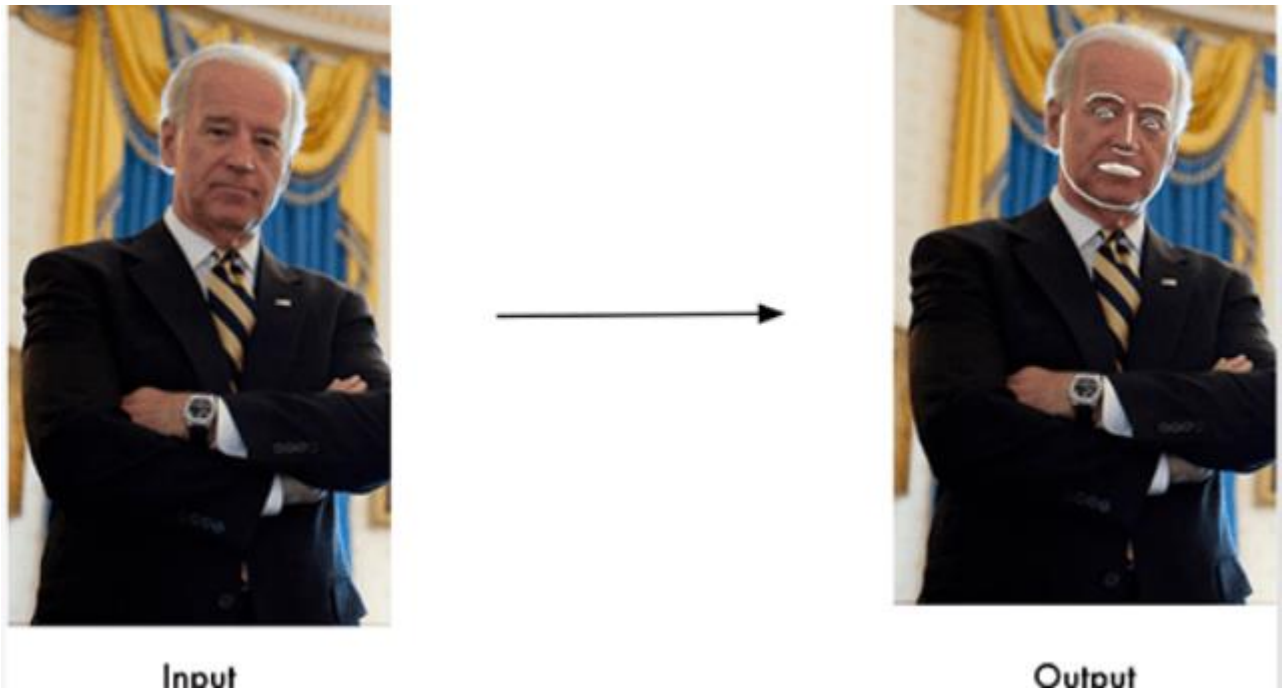


Рисунок 2.13 – Пошук рис обличчя на фотографіях

Карти координат конкретних точок грані. Контур очей складається з 6 точок, карта обличчя містить всього 68 точок, за допомогою яких можна з високою точністю порівнювати та ідентифікувати обличчя.

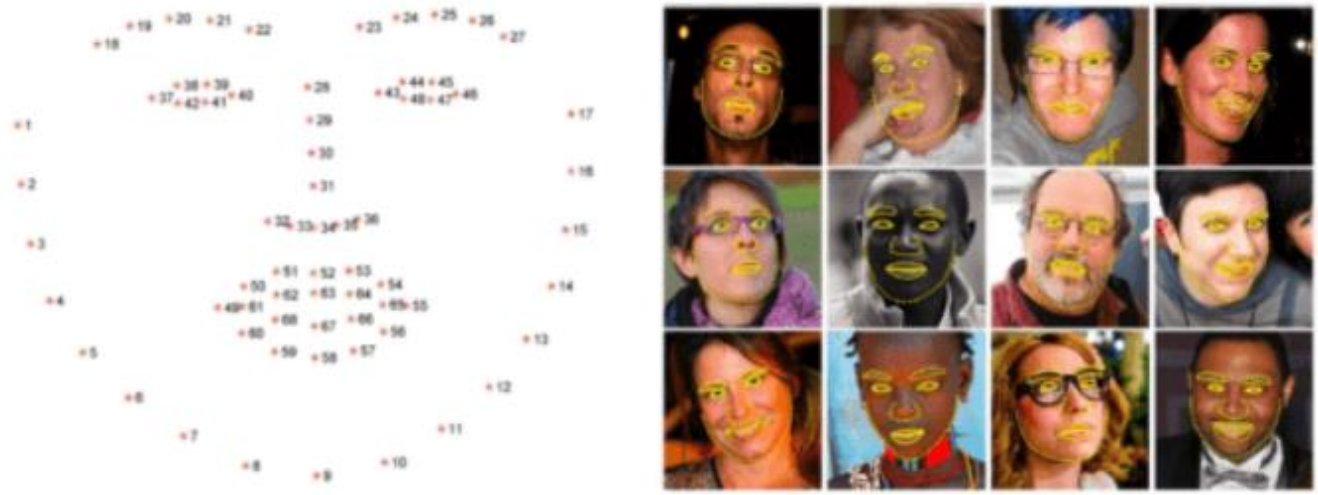


Рисунок 2.14 Карти координат специфічних лицьових точок

Ідентифікація облич на фотографіях. Отримайте дані про те, хто зображений на кожній фотографії. Можна створити базу співробітників чи клієнтів клубу,

розпізнавати їх і визначати «чужих» по співвідношеннюю координат точок обличчя.

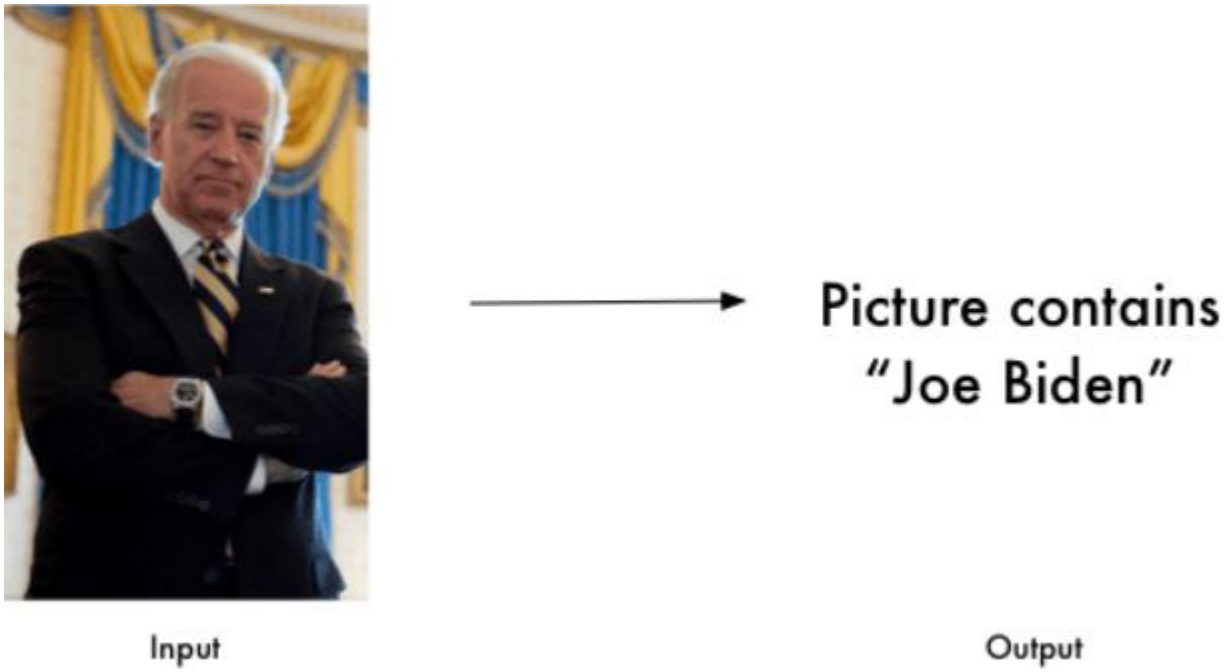


Рисунок 2.15 – Ідентифікація облич на фотографіях

Застосування в realtime-рішеннях. Також можливо застосовувати розпізнавання у роботі з потоковим відео та використовувати з іншими бібліотеками Python.

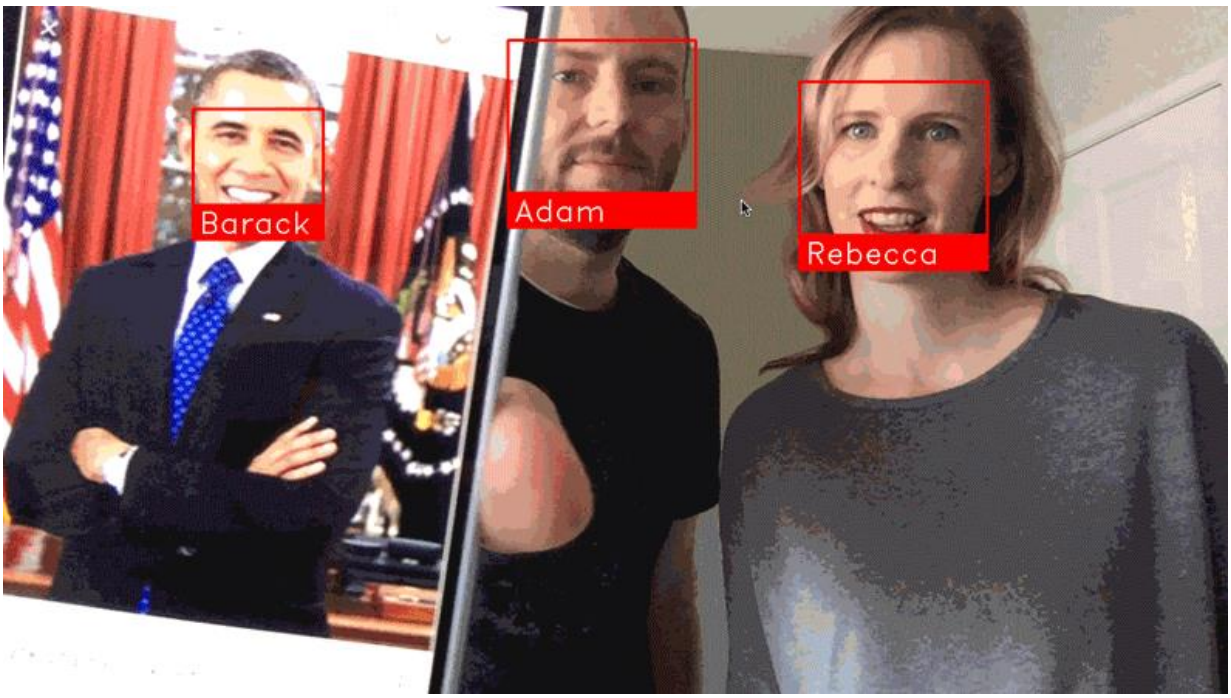


Рисунок 2.16 – Застосування в realtime-рішеннях

Для виконання задачі розпізнавання облич співробітників підприємства було вирішено використовувати TensorFlow і Keras. Так як, розпізнавання та класифікація зображень і є найпоширенішим застосуванням TensorFlow. Хоча методи описані вище працюють набагато швидше, ніж нейронні мережі з використанням TensorFlow, але для вирішення задач представлених у магістерській кваліфікаційній роботі доцільно буде використовувати саме TensorFlow, для синтезу й дослідження власної нейромережі з розпізнавання облич.

TensorFlow – це бібліотека з відкритим кодом, розроблена командою Google Brain під Python. TensorFlow компілює багато різних моделей та алгоритмів, що дозволяє користувачеві впроваджувати глибокі нейронні мережі для використання в таких задачах, як розпізнавання та класифікація зображень, а також обробки природної мови. TensorFlow – це потужна структура, яка функціонує завдяки реалізації серії вузлів для обробки, кожен з яких представляє собою математичні операції, а серія вузлів називається «графом».

Keras, це API високого рівня (інтерфейс програмування додатків), який використовує алгоритми TensorFlow, але не тільки він також може використовувати й інші бібліотеки ML. Keras розроблено з урахуванням зручності використання та модульності як керівних принципів. З точки зору використання Keras максимально спрощує реалізацію багатьох потужних, але дуже часто складних функцій TensorFlow, і його налаштовано для роботи з Python без значних змін чи налаштувань.

Розпізнавання зображень стосується задачі передачі зображення в нейронну мережу та призначення будь-якої помітки цьому зображенню. Помітка, виведена мережею, відповідатиме попередньо визначеному класу. Може призначатися відразу декілька класів, або ж тільки один. Якщо є лише один клас, зазвичай використовується термін «розпізнавання», тоді як задача розпізнавання декількох класів часто називають «класифікацією».

Підмножина класифікацій зображень уже є визначенням об’єкта, де певні екземпляри об’єктів розпізнаються як належні до певного класу, наприклад обличчя людини, номер автомобілю чи тварини.

Функція вилучення. Для того щоб виконувати операцію розпізнавання або класифікацію зображень, нейромережа повинна добути ознаки. Ознаки – це елементи даних зображення, які представляють значний інтерес і будуть передані через нейронну мережу. У конкретному випадку розпізнавання зображень ці ознаки є групами пікселів, таких як лінії та точки, які мережа аналізуватиме на наявність шаблонів.

Розпізнавання ознак (або виділення ознак) – це процес вилучення релевантних ознак із вхідного зображення для їх аналізу. Багато зображень містять анотації або метадані, які допомагають нейронній мережі знаходити релевантні ознаки.

Вилучення ознак із використанням фільтрів. Перший шар нейронної мережі приймає всі пікселі зображення. Після того, як усі дані подано в мережу, до зображення застосовуються різні фільтри, щоб зрозуміти різні частини зображення. Саме виділення ознак створює «карти ознак» [26].

Цей процес вилучення ознак із зображення виконується за допомогою використання «згорткового шару», а згортка вже формує представлення частини зображення. Саме з цього поняття згортки було отримано термін «Згортнова нейронна мережа» (CNN) – тип нейромережі, який найчастіше використовується для класифікації та розпізнавання зображень.

Ширина променю ліхтаря визначає розмір частини зображення, який переглядається за один раз, і в нейромережах є аналогічний параметр – саме розмір фільтра. Розмір фільтра впливає на кількість пікселів, які перевіряються одночасно. Загальний розмір фільтра, який використовує CNN, становить 3, і він охоплює як висоту, так і ширину, тому фільтром перевіряється область розміром 3 х 3 пікселя.

З цього зрозуміло, що для фільтра розміру 3, застосованого до повно колірного зображення, кінцеві розміри цього фільтра будуть 3 х 3 х 3. Для кожного

пікселя, який охоплює цей фільтр, мережа помножує значення фільтра на значення самих пікселів, для отримання числового значення цього пікселя . Далі та ж операція виконується для повного зображення, для того щоб отримати повне уявлення. Фільтр переміщається по всьому зображенню далі відповідно до параметра, який називається «крок», який визначає, на скільки пікселів потрібно перемістити фільтр після того, як він визначить значення на своїй поточній позиції. Типовим розміром для кроку CNN є два.

Остаточним результатом цих обчислень є карта ознак. Цей процес завжди виконується з використанням декількох фільтрів, для того щоб допомоги зберегти складність зображення.

Функція активації. Після створення карти ознак зображення значення, які представляють зображення, потрібно передати через активацію або шар активації. Функція активації приймає ці значення, які за допомогою згортковому шару знаходяться в лінійній формі (у формі списку чисел) і збільшує їх нелінійність, через те, що самі зображення не є лінійними [27].

Типовою функцією активації, яка використовується для цієї мети, є випрямлена лінійна одиниця (ReLU), хоча іноді також використовуються деякі інші функції активації.

Об'єднання шарів. Після активації дані надсилаються через з’єднувальний шар. Конкатенація «спрощує» зображення: воно отримує інформацію, яка представляє зображення, а потім стискає її. Процес об’єднання робить мережу більш гнучкою та здатною краще розпізнавати об’єкти та зображення на основі відповідних ознак [28].

Якщо подивитися на зображення, ми зазвичай не дбаємо про всю інформацію (наприклад, про те, що на фоні зображення), а лише про об’єкти, які нас цікавлять – машини, люди тощо.

Подібним чином шар об’єднання CNN позбавлятиметься від зайвих частин зображення, залишаючи лише ті частини, які він вважає потрібними, залежно від заданого розміру шару об’єднання.

Через те, що мережа повинна прийняти рішення щодо найважливіших частин зображення, розрахунок полягає в тому, що вона вивчатиме лише ті частини зображення, які насправді є суттю об’єкта, що розглядається. Це допомагає запобігти перенавчанню – коли мережа дуже пильно вивчає всі аспекти навчального прикладу і більше не може узагальнювати нові дані, через те, що враховує й нерелевантні відмінності.

Існують різні способи об’єднання значень, але найпоширенішим є максимальне поєднання. Максимальне поєднання означає взяття максимального значення серед пікселів в межах одного фільтра (тобто в межах однієї частини зображення). Це фільтрує три з чотирьох частини інформації при використанні фільтра 2 x 2 [29].

Максимальні значення пікселів потрібні для урахування можливих змін зображення, а кількість параметрів (розмір зображення) зменшується, щоб була можливість контролювати перетренування. Існують інші принципи поєднання, наприклад об’єднання середнього об’єму або об’єднання суми, але їх не використовують дуже часто, через те, що поєднання максимальних об’єднань дає більшу точність.

Стиснення. Остаточні шари CNN з розпізнавання зображення – тісно пов’язані шари – які вимагають представлення даних у векторній формі для обробки, яка буде далі. З цієї причини дані повинні бути «агреговані». Для цього значення стискаються в довгий вектор або стовпець послідовно впорядкованих чисел.

Повністю зв'язаний шар. Кінцевими шарами CNN є щільно з’єднані шари або штучні нейронні мережі (Artificial neural networks (ANN)). Основною функцією ШНМ є аналіз вхідних ознак і об’єднання їх у різні атрибути, які будуть допомагати у класифікації. Ці шари створюють набори нейронів, які представляють собою різні частини об’єктів, що розглядаються, і набір нейронів може представляти собою, наприклад, маленькі крильця комах або помаранчевий мандарин. Коли достатня

кількість цих нейронів запускається у відповідь на вхідне зображення, воно буде класифіковано як об’єкт.

Помилкою або різницею між обчисленими значеннями та очікуваним значенням у навчальному наборі обчислюється за допомогою ШНМ. Потім до мережі застосовується зворотнє поширення помилок, де обчислюється вплив даного нейрона на нейрон наступного шару, а потім його вплив (вага) виправляється. Це потрібно для оптимізації роботи моделі. Цей процес повторюється багато разів: таким чином мережа вчиться з даних і вивчає зв’язки між вхідними ознаками та вихідними класами.

Нейрони в середніх підключених шарах будуть виводити двійкові значення, які відносяться до можливих класів. Наприклад є три різних класи (кішка, птах, паркан), нейрон матиме значення «1» для класу, який, на його думку, представляє зображення, і значення «0» для інших класів.

Останній повністю зв'язаний шар, отримавши дані з попереднього шару, дає можливість кожному з класів всередині блоку (у сукупності). Якщо категорії кішки присвоєно значення 0,63, це означає, що ймовірність того, що зображення є кішкою, становить 63%.

2.3 Робочий процес машинного навчання

Для початку потрібно зібрати дані та впорядкувати їх так, щоб нейронна мережа могла з них навчатися. Це включає збір відповідних зображень і маркування їх. Навіть якщо завантажити датасет, підготовлений не вами, він, скоріш за все, потребуватиме певної попередньої обробки та підготовки, перш ніж ви зможете використати його для тренування. Підготовка даних сама по собі є складним процесом, який має справу з такими проблемами, як відсутні значення, пошкоджені дані, дані в невідомому форматі, хибні мітки тощо. У магістерській кваліфікаційній роботі використовувався попередньо оброблений набір даних [30].

Створення моделі. Створення моделі нейромережі передбачає вибір різних параметрів і гіперпараметрів. Потрібно визначитися з кількістю шарів, що будуть

використовуватися у моделі, якого розміру будуть вхідні та вихідні шари, які функції активації буде використано, чи будете ви використовувати Dropout [31].

Тренування моделі. Коли модель створено, залишається лише створити екземпляр моделі та адаптувати її до навчальних даних. Найбільша увага під час тренування моделі приділяється кількості часу, необхідного для навчання. Щоб задати тривалість тренування мережі, необхідно вказати кількість епох для тренування. Чим більше епох, тим більше навчається модель, і тим вище її ефективність, але якщо задати дуже багато епох навчання, є ризик перетренувати модель.

Оцінка моделі. Існує кілька етапів для визначення оцінки моделі. Спочатку потрібно порівняти роботу моделі з набором валідаційних даних: даними, на яких була натренована модель. Саме так, перевіряється робота моделі з новим набором даних і аналізується її продуктивність з використання різних показників.

Існують різні показники для визначення ефективності роботи моделі нейромережі, але самим поширеним є «точність», яка є відношенням правильно розпізнаних зображень до загальної кількості зображень у наборі даних.

Тому, як тільки буде отримано точність моделі в валідаційному наборі, доведеться повернутися та повторно до-тренувати мережу, використовуючи налаштовані параметри, через те, що малоймовірно, що мережа працюватиме добре під час першого навчання. Отже, потрібно продовжувати налаштовувати параметри мережі, перенавчати її та вимірювати продуктивність, доки не отримаєте задовільний результат точності мережі.

Наступним кроком є перевірка ефективності мережі вже на тестовому наборі даних. Це інший набір даних, з яким модель ніколи не працювала і не бачила його. Метою тестового набору є перевірка на наявність таких проблем, як перетренованість, щоб переконатися, що модель справді працює.

Висновки до розділу 2

В даному розділі була вирішена задача по вибору інструментальних засобів розробки системи з розпізнавання облич співробітників підприємства. Крім того, детально розглянуто робочий процес машинного навчання.

Було обрано мову програмування Python v.3.6 та спосіб розробки графічного інтерфейсу системи, а саме PyQt5. Середою розробки було обрано Python IDE під назвою PyCharm. Також було розглянуто методи, моделі та інформаційні технології для вирішення поставленої задачі, а саме було досліджено робочий процес машинного навчання, який включає у собі створення, навчання та оцінку моделі нейронної мережі. Для виконання задачі розпізнавання облич співробітників підприємства було вирішено використовувати TensorFlow і Keras. Так як, одним з найпоширеніших застосувань TensorFlow та Keras є розпізнавання та класифікація зображень.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ СПІВРОБІТНИКІВ ПІДПРИЄМСТВА

3.1 Розробка моделі нейронної мережі для розпізнавання облич

Сіамська нейромережа – один із найпростіших та найпопулярніших алгоритмів одноразового навчання. Методики, коли він кожного класу береться лише з одному навчальному прикладу. Таким чином, сіамська мережа зазвичай використовується у додатках, де в кожному класі є не так багато одиниць даних.

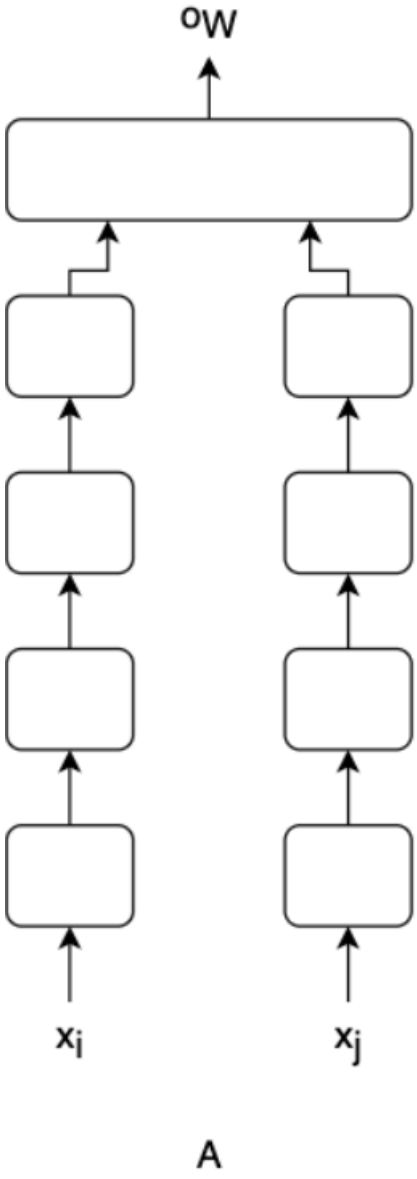


Рисунок 3.1 – Сіамська мережа

Припустимо, потрібно зробити модель розпізнавання осіб для підприємства, де працює близько 200 осіб. Якщо робити таку модель з нуля на основі згорткової нейромережі (CNN), то для навчання моделі та досягнення гарної точності розпізнавання нам знадобиться багато зображень кожного з цих 200 людей. Але очевидно, що такий датасет дуже складно зібрати, тому не варто робити модель на основі CNN або іншого алгоритму глибокого навчання, якщо немає достатньої кількості даних. У подібних випадках можна скористатися складним алгоритмом одноразового навчання, на кшталт сіамської мережі, яка може навчатися із використанням малій кількості даних.

По суті, сіамські мережі складаються з двох симетричних нейромереж, з однаковими вагами та архітектурою, які в кінці об'єднуються та використовують функцію енергії – E.

У магістерській кваліфікаційній роботі було розроблено систему розпізнавання облич співробітників підприємства із використанням створеної моделі розпізнавання облич на основі сіамських нейромереж.

Модель буде навчено визначати, коли дві особи однакові, а коли ні. Для цього спочатку скористаємося датасетом AT&T Database of Faces, який можна завантажити із сайту комп'ютерної лабораторії Кембриджського Університету.
Після завантаження і розпаковки можна побачити папки від s1 до s40 (рис. 3.1).

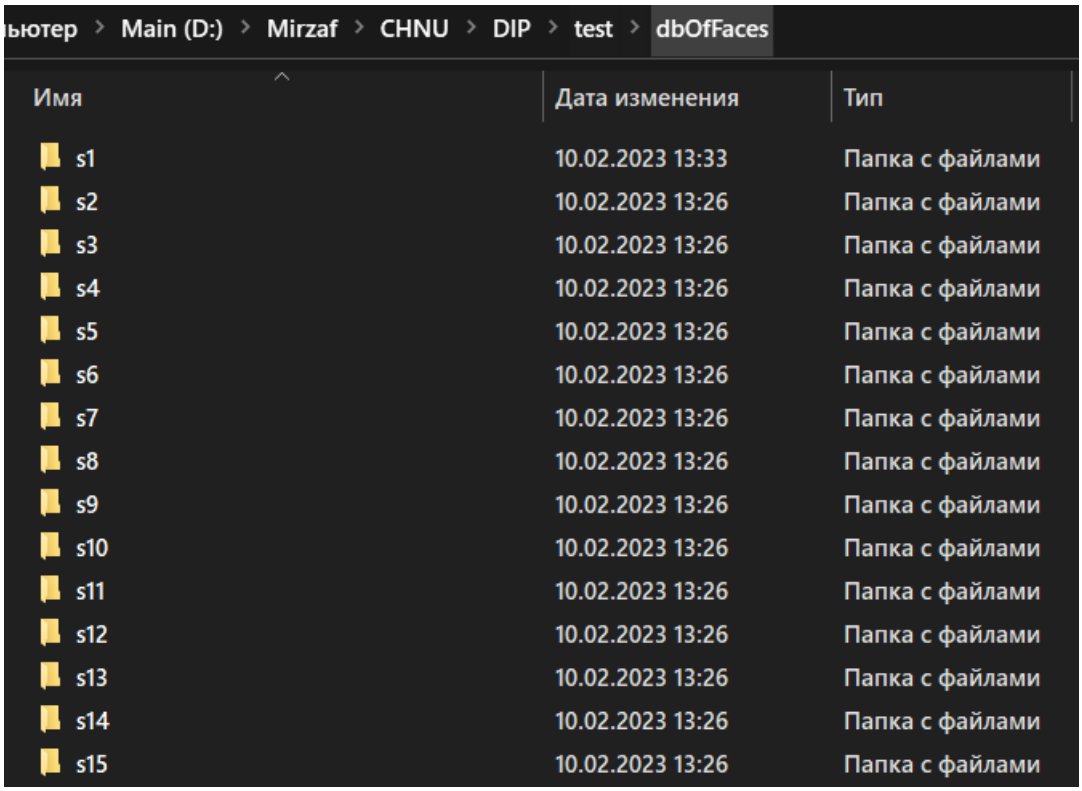


Рисунок 3.1 – Датасет AT&T Database of Faces

У кожній папці лежить 10 різних фотографій однієї людини, знятих з різних кутів. Ось вміст папки s1:

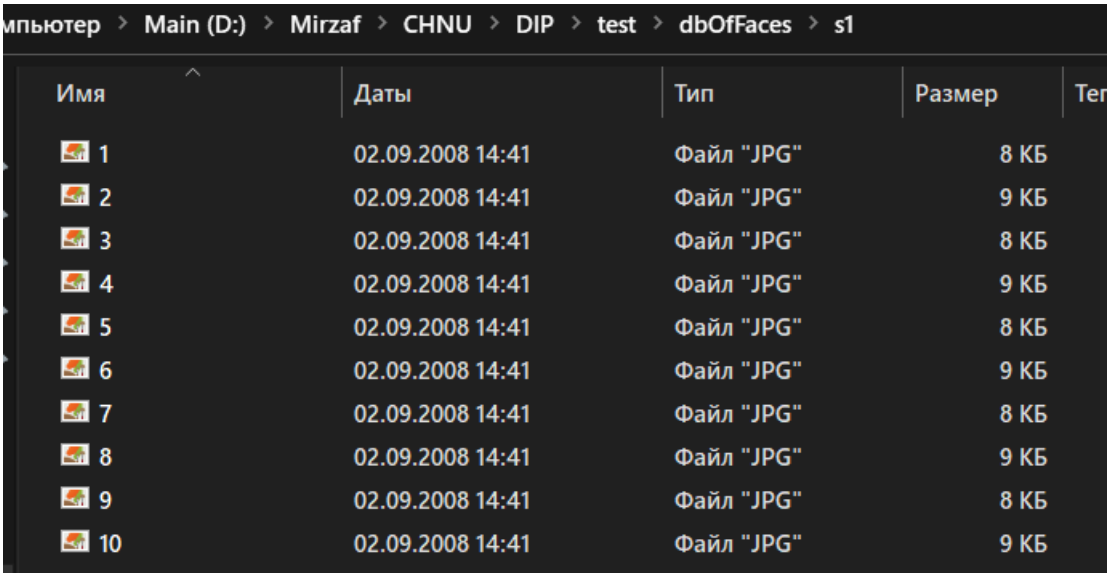
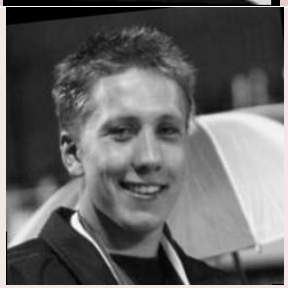


Рисунок 3.2 – Вміст папок датасету AT&T Database of Faces

Сіамським мережам потрібно подавати на вхід парні значення з маркуваннями, тому створимо такі набори. Візьмемо з однієї папки дві випадкові фотографії і позначимо як справжню пару (genuine). Потім візьмемо дві фотографії з різних папок і позначимо як «хибну» пару (opposite):

Таблиця 3.1 – Приклад створення наборів фотографій

Input pair	Label
 	**Genuine**
 	**Opposite**
 	**Genuine**
 	**Opposite**

Розподіливши всі фотографії по маркованих парах, візьмемося за навчання мережі. З кожної пари ми одну фотографію передаємо мережі А, а другу — мережі Б. Обидві мережі лише витягують вектори властивостей. Для цього скористаємось двома згортковими шарами з активаціями блоків лінійної ректифікації (rectified linear unit (ReLU)). Вивчивши властивості, ми передаємо згенеровані обома

мережами вектори в функцію енергії, що оцінює схожість. Як функцію скористаємося евклідовою відстанню.

Для початку роботи потрібно встановити бібліотеки з якими будемо працювати.

Використовуючи Python IDE під назвою PyCharm, це дуже легко зробити через вбудований термінал. Для цього потрібно ввести команду: «pip install tensorflow», як зображено на рис. 3.3.

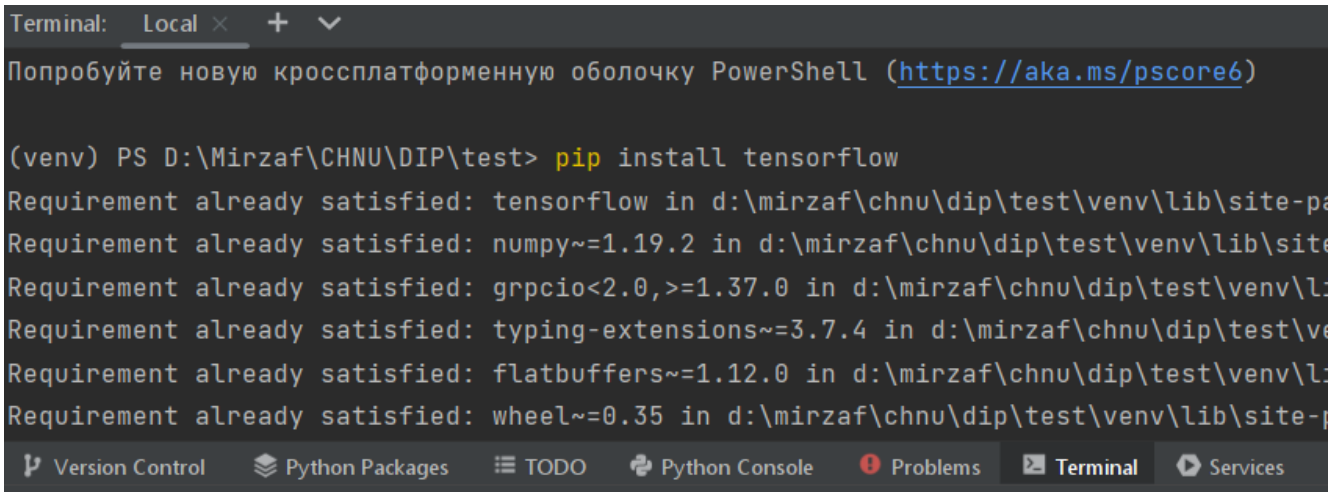


Рисунок 3.3 – Встановлення TensorFlow

Таким чином встановлюються також і всі інші потрібні для роботи бібліотеки. Далі потрібно імпортувати необхідні бібліотеки (рис. 3.4).

```
import re
import numpy as np
from PIL import Image
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import tensorflow as tf
from tensorflow import keras

from sklearn.model_selection import train_test_split
from keras import backend as K
from keras.layers import Activation
from keras.layers import Input, Lambda, Dense, Dropout, Convolution2D, MaxPooling2D, Flatten
from keras.models import Sequential, Model
from keras.optimizers import RMSprop
```

Рисунок 3.4 – Імпортування бібліотек

Тепер визначимо функцію читання вхідних зображень. Функція read_image бере картинку і повертає NumPy-масив. Спочатку ми читаємо зображення як необроблений файл у буфер. За допомогою регулярного виразу ми витягуємо заголовок, ширину, висоту та максимальне значення зображення. Потім ми перетворюємо зображення в масив numpy за допомогою np.frombuffer, який інтерпретує буфер як одновимірний масив (рис. 3.5).

```
def read_image(filename, byteorder='>'):
    # спочатку ми читаємо зображення як необроблений файл у б
    with open(filename, 'rb') as f:
        buffer = f.read()

    # за допомогою регулярного виразу ми витягуємо заголовок,
    header, width, height, maxval = re.search(
        b"(^P5\s(?:\s*#.*[\r\n])*"
        b"(\d+)\s(?:\s*#.*[\r\n])*"
        b"(\d+)\s(?:\s*#.*[\r\n])*"
        b"(\d+)\s(?:\s*#.*[\r\n]\s)*)", buffer).groups()

    # потім ми перетворюємо зображення в масив numpy за допом
    return np.frombuffer(buffer, dtype='u1' if int(maxval)
```

Рисунок 3.5 – Функція читання вхідних зображень

Тепер визначимо функцію get_data, яка генеруватиме дані. Нагадаю, що сіамським мережам потрібно подавати пари даних (genuine та imposite) з двійковим маркуванням.

Спочатку прочитаємо зображення (img1, img2) з однієї директорії, збережемо їх у масиві x_genuine_pair, надаємо y_genuine значення 1. Потім прочитаємо зображення (img1, img2) з різних директорій, збережемо їх у парі x_imposite, і надаємо y_imposite значення 0.

Конкатенуємо x_genuine_pair і x_opposite в X, а y_genuine і y_opposite - в Y (Рисунок 3.6)

```python
def get_data(size, total_sample_size):
    image = read_image('dbOfFaces/s' + str(1) + '/' + str(1) + '.pgm', 'rw+')
    image = image[::size, ::size]
    dim1 = image.shape[0]
    dim2 = image.shape[1]
    count = 0
    x_geuine_pair = np.zeros([total_sample_size, 2, 1, dim1, dim2])  # 2 is for pairs
    y_genuine = np.zeros([total_sample_size, 1])
    for i in range(40):
        for j in range(int(total_sample_size / 40)):
            ind1 = 0
            ind2 = 0
            while ind1 == ind2:
                ind1 = np.random.randint(10)
                ind2 = np.random.randint(10)
            img1 = read_image('dbOfFaces/s' + str(i + 1) + '/' + str(ind1 + 1) + '.pgm', 'rw+')
            img2 = read_image('dbOfFaces/s' + str(i + 1) + '/' + str(ind2 + 1) + '.pgm', 'rw+')
            img1 = img1[::size, ::size]
            img2 = img2[::size, ::size]
            x_geuine_pair[count, 0, 0, :, :] = img1
            x_geuine_pair[count, 1, 0, :, :] = img2
            y_genuine[count] = 1
            count += 1
    count = 0
    x_imposite_pair = np.zeros([total_sample_size, 2, 1, dim1, dim2])
    y_imposite = np.zeros([total_sample_size, 1])
```

Рисунок 3.6 – Функція get_data, яка генеруватиме дані

```python
    for i in range(int(total_sample_size / 10)):
        for j in range(10):

            while True:
                ind1 = np.random.randint(40)
                ind2 = np.random.randint(40)
                if ind1 != ind2:
                    break

            img1 = read_image('dbOfFaces/s' + str(ind1 + 1) + '/' + str(j + 1) + '.pgm', 'rw+')
            img2 = read_image('dbOfFaces/s' + str(ind2 + 1) + '/' + str(j + 1) + '.pgm', 'rw+')

            img1 = img1[::size, ::size]
            img2 = img2[::size, ::size]

            x_imposite_pair[count, 0, 0, :, :] = img1
            x_imposite_pair[count, 1, 0, :, :] = img2

            y_imposite[count] = 0
            count += 1
    X = np.concatenate([x_geuine_pair, x_imposite_pair], axis=0) / 255
    Y = np.concatenate([y_genuine, y_imposite], axis=0)

    return X, Y
```

Рисунок 3.7 – Функція get_data, яка генеруватиме дані

Тепер згенеруємо дані та перевіримо їх розмір. У нас 20 000 фотографій, з яких зібрано 10 000 справжніх та 10 000 помилкових пар:

```
X, Y = getdata(_size, _total_samples_sizes_)
X._shapes
(19990, 3, 2, 46, 36)
Y._shapes
(19990, 2)
```

Розділимо весь масив інформації: 75% пар піде на навчання, а 25% - на тестування:

x_trains_, x_tests_, y_trains_, y_tests_ =_ trains_tests_splits_(X, Y, _test_sizes=.25)

Тепер створимо сіамську мережу. Спочатку визначимо базову мережу – це буде згорткова нейромережа для отримання властивостей (Рисунок 3.8). Створимо два згорткові шари за допомогою ReLU-активацій та шару з визначенням максимального значення (max pooling) після flat-шару:

```
100  def build_base_network(input_shape):
101      seq = Sequential()
102
103      nb_filter = [6, 12]
104      kernel_size = 3
105
106      # згортковий шар 1
107      seq.add(Convolution2D(nb_filter[0], kernel_size, kernel_size, input_shape=input_shape,
108                            border_mode='valid', dim_ordering='th'))
109      seq.add(Activation('relu'))
110      seq.add(MaxPooling2D(pool_size=(2, 2)))
111      seq.add(Dropout(.25))
112
113      # згортковий шар 2
114      seq.add(Convolution2D(nb_filter[1], kernel_size, kernel_size, border_mode='valid', dim_ordering='th'))
115      seq.add(Activation('relu'))
116      seq.add(MaxPooling2D(pool_size=(2, 2), dim_ordering='th'))
117      seq.add(Dropout(.25))
118
119      # flat-шар
120      seq.add(Flatten())
121      seq.add(Dense(128, activation='relu'))
122      seq.add(Dropout(0.1))
123      seq.add(Dense(50, activation='relu'))
124      return seq
```

Рисунок 3.8 – Створення базової мережі

Потім передаємо пару зображень базовій мережі, яка поверне векторні уявлення, тобто вектори властивостей:

```
input_dimaD = xs_train.shapes_[3:]
```

```
image_a = Input(shape=input_dim)
image_b = Input(shape=input_dim)
base_network_ = builds_bases_networks(input_dimaD)
feats_vectors_a = base_network_(image_a)
feats_vectors_b = base_network_(image_b)
```

feats_vectors_a та feats_vectors_b – це вектори властивостей пари зображень. Далі передаємо їх функції енергії для обчислення дистанції між ними. А як функцією енергії скористаємося евклідовою відстанню:

```
134  def euclidean_distance(vects):
135      x, y = vects
136      return K.sqrt(K.sum(K.square(x - y), axis=1, keepdims=True))
137
138
139  def eucl_dist_output_shape(shapes):
140      shape1, shape2 = shapes
141      return (shape1[0], 1)
142
143  distance = Lambda(euclidean_distance, output_shape=eucl_dist_output_shape)([feat_vecs_a, feat_vecs_b])
```

Рисунок 3.9 – Функція енергії

Задаємо число епох рівним 13, застосуємо властивість RMS для оптимізації та оголосимо модель.

```
145  epochs = 13
146  rms = RMSprop()
147  model = Model(input=[input_a, input_b], output=distance)
```

Рисунок 3.10 – Оголошення моделі

Тепер визначимо функцію втрат contrastive_loss function і скомпілюємо модель.

```
149  def contrastive_loss(y_true, y_pred):
150      margin = 1
151      return K.mean(y_true * K.square(y_pred) + (1 - y_true) * K.square(K.maximum(margin - y_pred, 0)))
152
153  model.compile(loss=contrastive_loss, optimizer=rms)
```

Рисунок 3.11 – Функція втрат

Далі можна запустити навчанням моделі впродовж 13 епох (Рисунок 3.12)

```
155  img_1 = x_train[:, 0]
156  img_2 = x_train[:, 1]
157
158  model.fit([img_1, img_2], y_train, validation_split=.25, batch_size=128, verbose=2, nb_epoch=epochs)
159
```

Рисунок 3.12 – Навчання моделі

В результаті маємо, що по мірі проходження епох моделлю знижаються втрати.

```
D:\Mirzaf\CHNU\DIP\test\venv\Scripts\python.exe D
Train on 11250 samples, validate on 3750 samples
Epoch 1/13
 - 60s - loss: 0.2179 - val_loss: 0.2156
Epoch 2/13
 - 53s - loss: 0.1520 - val_loss: 0.2102
Epoch 3/13
 - 53s - loss: 0.1190 - val_loss: 0.1545
Epoch 4/13
 - 55s - loss: 0.0959 - val_loss: 0.1705
Epoch 5/13
 - 52s - loss: 0.0801 - val_loss: 0.1181
Epoch 6/13
 - 52s - loss: 0.0684 - val_loss: 0.0821
Epoch 7/13
 - 52s - loss: 0.0591 - val_loss: 0.0762
Epoch 8/13
 - 52s - loss: 0.0526 - val_loss: 0.0655
Epoch 9/13
 - 52s - loss: 0.0475 - val_loss: 0.0662
Epoch 10/13
 - 52s - loss: 0.0444 - val_loss: 0.0469
Epoch 11/13
 - 52s - loss: 0.0408 - val_loss: 0.0478
Epoch 12/13
 - 52s - loss: 0.0381 - val_loss: 0.0498
Epoch 13/13
- 54s - loss: 0.0356 - val_loss: 0.0363
```

Рисунок 3.13 – Проходження епох

Після цього можна перевірити роботу моделі на тестових даних, та визначимо функцію для обчислення точності.

```
pred = model.predict([x_test[:, 0], x_test[:, 1]])
def compute_accuracy(predictions, labels):
    return labels[predictions.ravel()
```

Рисунок 3.14 – Перевірка роботи моделі

У результаті визову функції для обчислення точності було отримано показник точності, який дорівнює 93% (рис. 3.15).

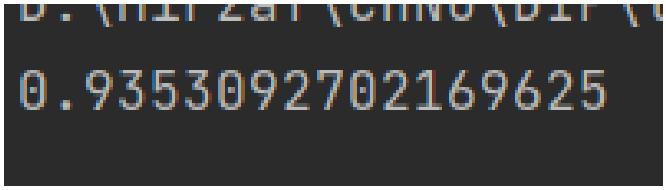


Рисунок 3.15 – результаті визову функції для обчислення точності

3.2 Розробка інтерфейсу програми

За допомогою бібліотеки OpenCV, було отримано зображення з камери та використано навчену модель «test_model.xml» для розпізнавання обличчя.

Система опрацьовує зображення з камери для визначення чи є людина співробітником підприємства. Отже для доступу співробітник має натиснути на кнопку «Отримати доступ» (рис. 3.17).

Якщо система знайде схожість отриманого зображення людини з камери із зображенням в базі співробітників – працівник підприємства зможе увійти. В базу даних буде записано відеозапис, дату та час запиту доступу.

У вікні програми є дві вкладки «Ідентифікація» та «Записи».

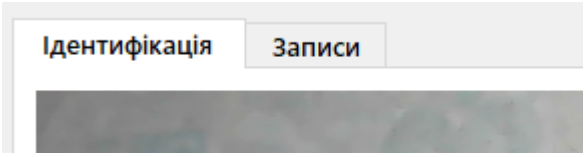


Рисунок 3.16 – Вкладки вікна програми

На першій вкладці здійснюється ідентифікація працівника. У лівій частині вікна транслюється відео з камери на пропускній (на рис. 3.17 для прикладу транслюється зображення з веб-камери) та кнопка «Отримати доступ». В правій частині вікна виводиться результат ідентифікації.

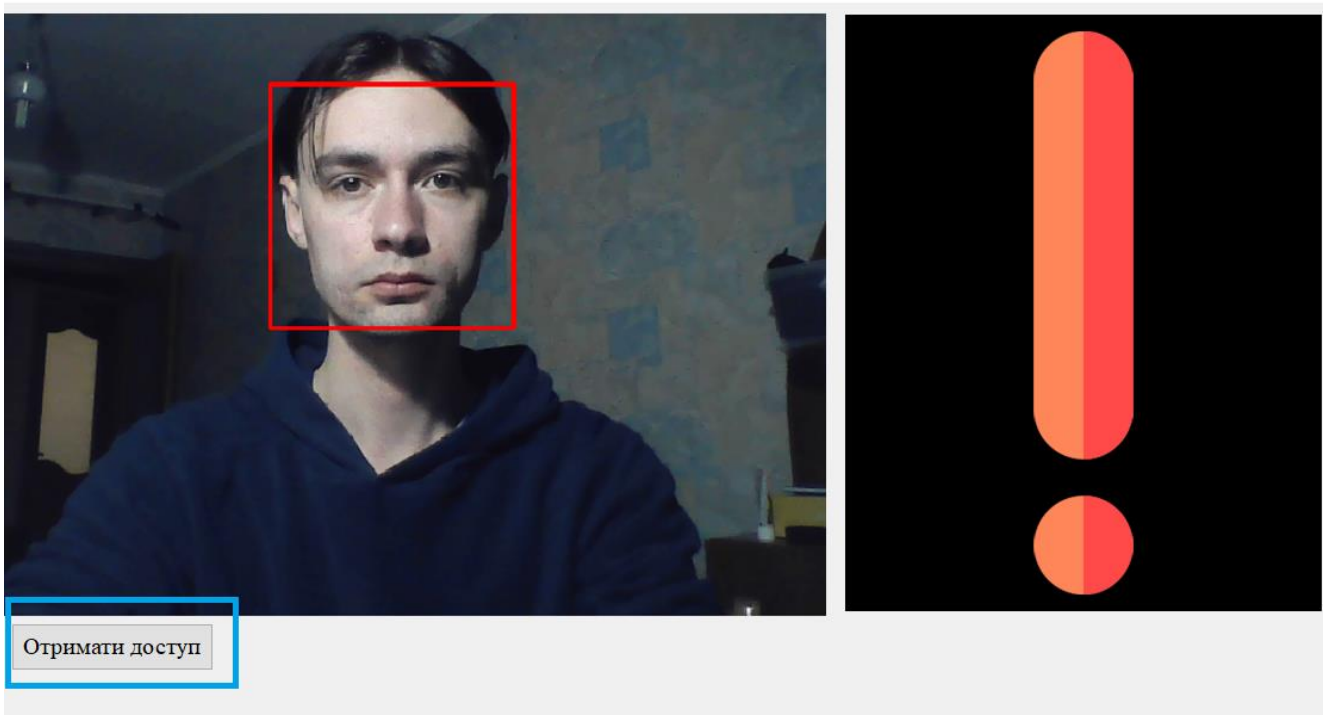


Рисунок 3.17 – Кнопка отримати доступ

Після натискання на кнопку «Отримати доступ» система використовує навчену модель для порівняння зображення з веб-камери та тими зображеннями які є в базі співробітників. Якщо системі вдається ідентифікувати працівника – вона надає доступ, виводиться фотографія працівника та реєструється дата і час запиту доступу. Також для додаткової безпеки після натискання на кнопку «Отримати доступ» виконується запис зображення у вигляді відео тривалістю 10 секунд. Запис потрібен для того, щоб у випадку порушень чи поганих інцидентів, наприклад крадіжки майна співробітників або підприємства, можна було переглянути записи отримання доступу співробітників, та побачити намагалися когось провести на підприємство, тощо.

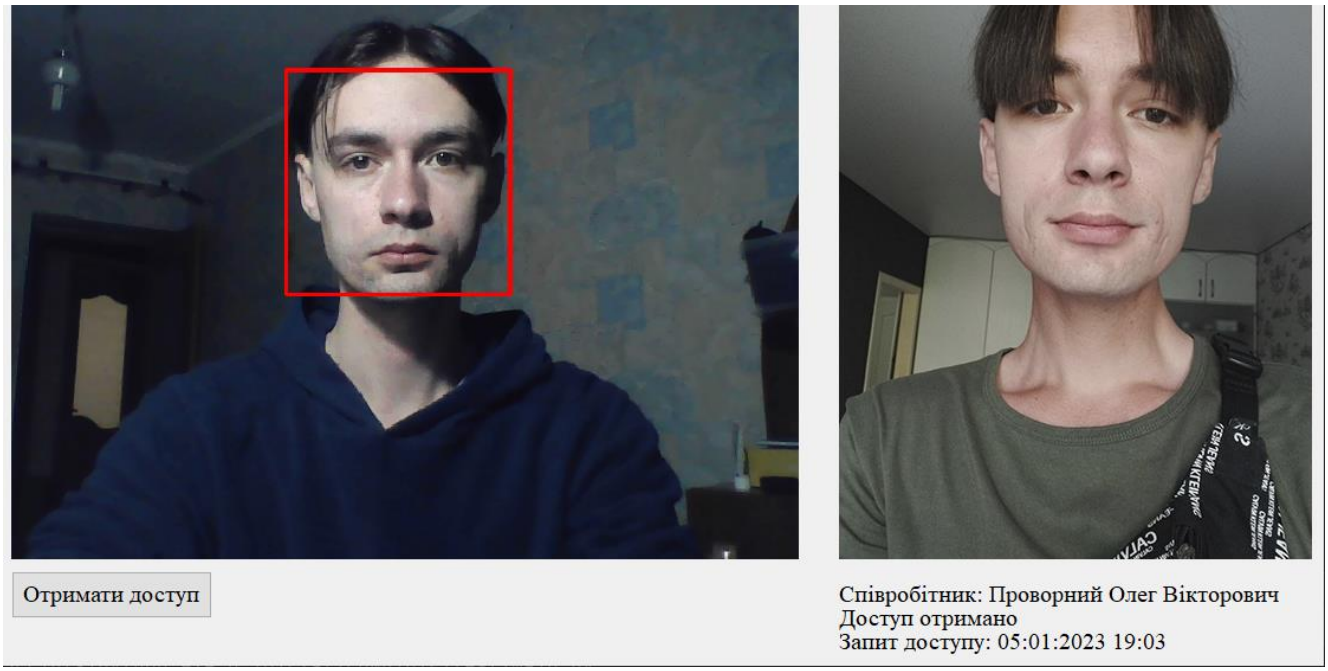


Рисунок 3.18 – Результат роботи програми

Переглянути записи отримання доступу можна на другій вкладці програми «Записи» (рис. 3.19).

Рисунок 3.19 – Вкладка «Записи»

На цій вкладці вікно складається з плеєру для перегляду записів, та двох кнопок «Відкрити» і «Play/Pause». За допомогою кнопки «Відкрити» можна обрати файл з директорії на сервері куди зберігаються відеозаписи доступу співробітників підприємства (рис. 3.20).

Рисунок 3.20 – Кнопка «Відкрити»

Кнопка «Play/Pause» відповідає за запуск обранного з директорії на сервері відеозапису доступу співробітників підприємства (рис. 3.21).

Рисунок 3.21 – Кнопка «Play/Pause»

Відкривши потрібний відеозапис та натиснувши кнопку «Play» можна переглянути запис отримання доступу з ідентифікованим на ньому працівником (рис. 3.22).

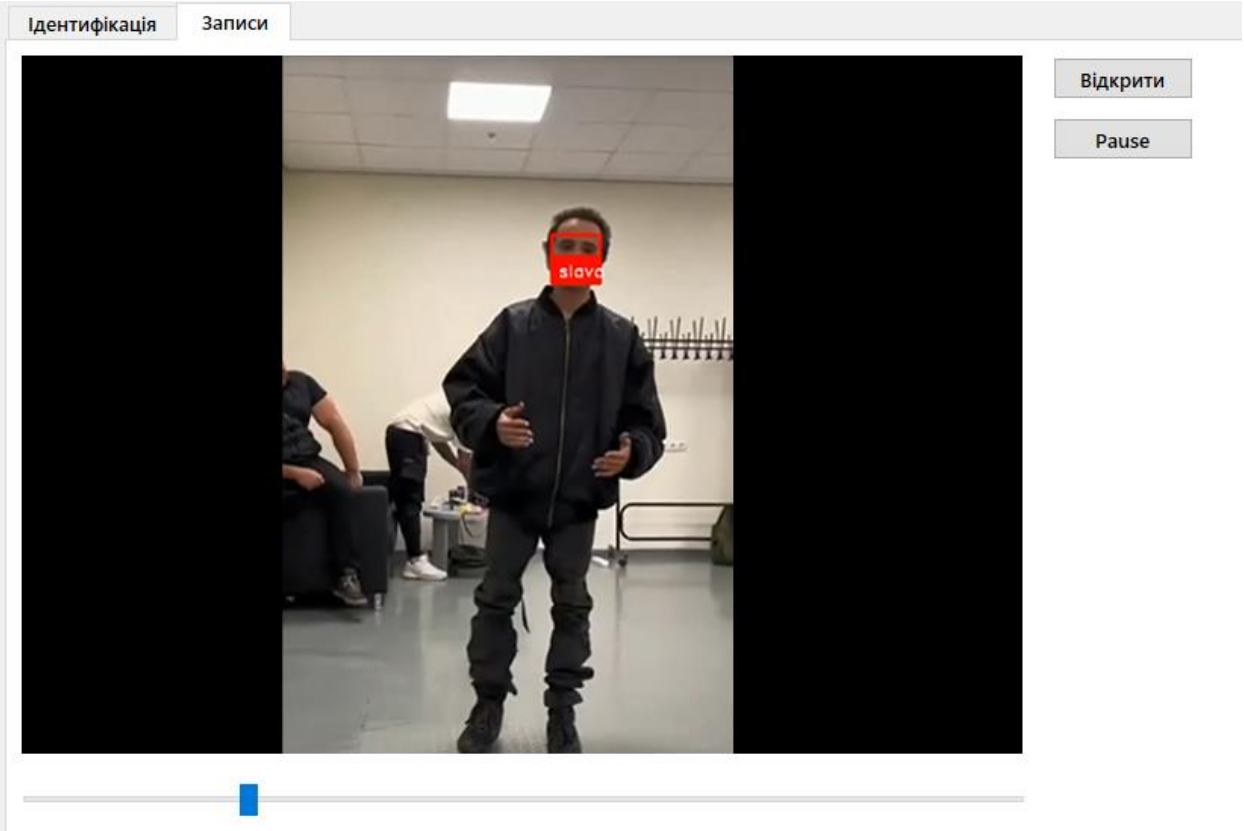


Рисунок 3.22 – Перегляд відеозапису

Висновки до розділу 3

В даному розділі у результаті роботи над розроблюваною програмою було створено систему для розпізнавання співробітників підприємства використовуючи мову Python та такі бібліотеки як TensorFlow/Keras, OpenCV, Numpy.

Створена система працює на базі Сіамської мережі, яка складається з базової згорткової нейромережі для отримання властивостей, двох згорткових шарів з функцією активації ReLU та шару з визначенням максимального значення після flat-шару. Графічний інтерфейс програми розроблено з використанням PyQt5. Вікно програми складається із двох вкладок «Ідентифікація» та «Записи». На першій вкладці проводиться ідентифікація співробітників підприємства та надається доступ, якщо результат позитивний. Також реєструється дата та час отримання доступу, і робиться відеозапис запиту. На другій вкладці можна переглянути отримані після ідентифікації співробітників відеозаписи їх запитів доступу. Розроблена система надає можливість ідентифікувати співробітників підприємства.

ВИСНОВКИ

У магістерській кваліфікаційній були розглянуті методи, моделі та програмні засоби розпізнавання. Було проведено огляд сучасних технологій для створення систем розпізнавання облич. Також проаналізовані наявні аналоги систем управління та контролю доступу. Проведено огляд та аналіз інструментальних засобів розробки. В результаті на основі представленого теоретичного матеріалу були обрані інструментальні методи та засоби розробки. Була розроблена програма за допомогою якої можна ідентифікувати співробітників підприємства. Таким чином було підвищено ефективність системи контролю доступу співробітників підприємства за допомогою використання штучних нейронних мереж для розпізнавання облич.

В першому розділі було розглянуте поняття штучних нейронних мереж та процес розпізнавання облич. Крім того детально проаналізовано наявні системи розпізнавання облич, які є лідерами у світі. З кожним роком зростання інтересу до технологій розпізнавання облич помітно зростає. На сьогоднішній день вже існує кілька десятків систем розпізнавання, які застосовують у різних сферах людського життя.

В другому розділі була вирішена задача по вибору інструментальних засобів розробки системи з розпізнавання облич співробітників підприємства. Крім того, детально розглянуто робочий процес машинного навчання. Було обрано мову програмування Python v.3.6 та спосіб розробки графічного інтерфейсу системи, а саме PyQt5. Середою розробки було обрано Python IDE під назвою PyCharm. Також було розглянуто методи, моделі та інформаційні технології для вирішення поставленої задачі, а саме було досліджено робочий процес машинного навчання, який включає у собі створення, навчання та оцінку моделі нейронної мережі. Для виконання задачі розпізнавання облич співробітників підприємства було вирішено використовувати TensorFlow і Keras. Так як, одним з найпоширеніших застосувань TensorFlow та Keras є розпізнавання та класифікація зображень.

В третьому розділі у результаті роботи над розроблюваною програмою було створено систему для розпізнавання співробітників підприємства використовуючи мову Python та такі бібліотеки як TensorFlow/Keras, OpenCV, Numpy.

Створена система працює на базі Сіамської мережі, яка складається з базової згорткової нейромережі для отримання властивостей, двох згорткових шарів з функцією активації ReLU та шару з визначенням максимального значення після flat-шару. Графічний інтерфейс програми розроблено з використанням PyQt5. Вікно програми складається із двох вкладок «Ідентифікація» та «Записи». На першій вкладці проводиться ідентифікація співробітників підприємства та надається доступ, якщо результат позитивний. Також реєструється дата та час отримання доступу, і робиться відеозапис запиту. На другій вкладці можна переглянути отримані після ідентифікації співробітників відеозаписи їх запитів доступу. Розроблена система надає можливість ідентифікувати співробітників підприємства.

В результаті магістерської кваліфікаційної роботи було спроектовано систему розпізнавання облич співробітників підприємства у відеопотоках на основі за допомогою штучних нейронних мереж. Було розроблено модель для розпізнавання облич на основі сіамських мереж.

Розроблена система може використовуватися при розв'язання різних завдань відео аналітики і, в першу чергу, має безпосереднє застосування в системах контролю доступу та ідентифікації особистості.

В ході дослідження були вирішені завдання:

Проведено аналіз існуючих на ринку видів інтелектуальних систем розпізнавання облич; визначено основні інструментальні засоби для розробки системи; спроектовано систему розпізнавання облич співробітників підприємства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lopatina OL, Komleva YK, Gorina YV, Higashida H and Salmina AB. Neurobiological Aspects of Face Recognition: The Role of Oxytocin. *Front. Behav. Neurosci.* 12:195. 2018 URL: https://www.frontiersin.org/articles/10.3389/fnbeh.2018.00195/full
2. Kinnikar A., Husain M., Meena S.M. Face Recognition Using Gabor Filter And Convolutional Neural Network // Proceedings of the International Conference on Informatics and Analytics (Pondicherry, India, August 25–26, 2016), 2016. P. 113:1–113:4. DOI: 10.1145.Khandelwal R. Convolutional Neural Network (CNN) Simplified. 2018.
3. Facial Emotion Recognition Using Deep Convolutional Neural Network / E. Pranav; Suraj Kamal; C. Satheesh Chandran; M.H. Supriya, 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS).
4. Шитиков В.К., Розенберг Г.С., Зинченко Т.Д. Методы системной идентификации / Тольятти: ИЭВБ РАН. 2003. 463 с.
5. Крисилов В.А. Представление исходных данных в задачах нейросетевого программирования / Одесса: ОНПУ. 2003.
6. Введение в теорию нейронных сетей [Электронный ресурс]//Основные понятиянейросетей/ URL: http://www.orc.ru (дата обращения 12.04.2017)
7. Шахнов В.А., Власов А.И., Кузнецов А.С. Нейрокомпьютеры: архитектура исхемотехника / М.: Изд-во Машиностроение. 2000. 64 с.
8. Хайкин С. Нейронные сети. Полный курс / М.: Вильямс, 2006.
9. Халафян А.А. Статистический анализ данных. 3-е изд. учеб./ Бином – Пресс.2007. 512 с.
10. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем / СПб: Наука и техника. 2005. 384 с.
11. Остроух А.В. Интеллектуальные системы Учебное пособие/ Научно-инновационный центр. 2015 г.

12. Рассел С., Норвиг П. Искусственный интеллект: современный подход / Изд-во: Вильямс. 2006. 1424 с.
13. Системы технической безопасности и охраны «MTI». FaceVACS – VideoScan [Электронный ресурс]. // Режим доступа: http://www.security.mti.ua/products/sistemy-videonabludeniya/Soft-raspoznavanie-liz/Cognitec/152-facevacsvideoscan_/ .
14. NEC. Facial Recognition [Электронный ресурс]. // Режим доступа: https://ru.nec.com/solutions/security/technologies/face_recognition.html.
15. VisionLabs. LUNA SDK [Электронный ресурс]. // Режим доступа: https://visionlabs.ai/ru/luna-platform-info.html .
16. Neurotechnology. VeriLook SDK [Электронный ресурс]. // Режим доступа: http://www.neurotechnology.com/ .
17. Python [Електронний ресурс] – Режим доступу: https://www.python.org/about/
18. PyCharm [Електронний ресурс] – Режим доступу: https://www.jetbrains.com/ru-ru/pycharm/
19. Spyder [Електронний ресурс] – Режим доступу: https://www.spyder-ide.org
20. Atom [Електронний ресурс] – Режим доступу: https://github.blog/2022-06-08-sunsetting-atom/
21. PyQt [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/PyQt
22. Qt [Електронний ресурс] – Режим доступу: https://www.qt.io
23. Qt Designer [Електронний ресурс] – Режим доступу: https://doc.qt.io/qt-6/qtdesigner-manual.html
24. OpenCV [Електронний ресурс] – Режим доступу: https://opencv.org
25. Ніколаєв, С. С. Залежність якості детектора облич на ознаках Хаара від варіативності навчальної вибірки / С. С. Ніколаєв, Ю. О. Тимошенко, К. Ю. Матвіїв // Наукові вісті НТУУ «КПІ» : міжнародний науково-технічний журнал. – 2017. – № 6(116). – С. 38–46. – Бібліогр.: 6 назв.

26. Головко В.А. Нейронные сети: обучения, организация и применение / М.: ИПРЖР. 2008
27. Гундырев К.В. Искусственные нейронные сети в задачах диагностирования рельсовых цепей// Науч.-иссл.лабор. «Компьютерные системы автоматики». 2005.
28. Заенцев И.В. Нейронные сети: основные модели/Учебное пособие физ. ф-та. Воронеж. 1999.
29. Иванов А.И. Нейросетевые алгоритмы биометрической идентификации / Изд-во: Радиотехника. 2006. 144 с.
30. Кочеткова А.С. Применение нейронных сетей для мониторинга безопасности //Серия 9: Исследования молодых ученых. 2007.
31. Крисилов В.А. Представление исходных данных в задачах нейросетевого программирования / Одесса: ОНПУ. 2003
32. Гетия И.Г., Леонтьева И.Н., Шумилин В.К. Методические указания по проведению занятия по дисциплине «Безопасность жизнедеятельности » на тему: «Определение интегральной бальной оценки тяжести труда на робочем месте» . – М.: МГАПИ, 2002. – 22 с.
33. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу //Охорона праці. – 2001. –№ 12. – С. 12-20.
34. Жидецький В.Ц. Основи охорони праці. Підручник. – Львів: УАД, 2006. – 336 с.
35. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.