

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова**

Навчально-науковий інститут автоматики та електротехніки

(повне найменування інституту, факультету)

Кафедра комп'ютеризованих систем управління

(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри

_____ 
« ____ » _____ 2024 р.

Пояснювальна записка

до кваліфікаційної роботи

на здобуття другого (магістерського) рівня вищої освіти

**за спеціальністю 174 – «Автоматизація, комп'ютерно-інтегровані
технології та робототехніка»**

ОПП - «Комп'ютеризовані системи управління та автоматика»

на тему: **Нейромережева система ідентифікації особи на підприємстві**

Виконав студент **6** курсу, **6341м** групи

Проворний І.В. 

(прізвище та ініціали)

Керівник

Герасін О. С. 

(прізвище та ініціали)

Національний університет кораблебудування імені адмірала Макарова

ННІ	автоматики та електротехніки
Кафедра	комп'ютеризованих систем управління
Рівень вищої освіти	другий (магістерський)
Спеціальність	174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»
ОПП	«Комп'ютеризовані системи управління і автоматика»

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

 Черно О.О.
« ____ » _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Пр о в о р н и й І г о р В і к т о р о в и ч

(прізвище, ім'я, по батькові)

1. Тема роботи: нейромережева система ідентифікації особи на підприємстві

керівник роботи Герасін Олександр Сергійович, к.т.н., доцент

затверджені наказом вищого навчального закладу від "14" жовтня 2024 р. № 1075-уч

2. Строк подання роботи 18.12.2024 р.

3. Вихідні дані до роботи при розробці системи задіяти нейромережеві технології.
Забезпечити можливість роботи системи з зображеннями та відеопотоком. Точність
розпізнавання осіб має становити не менше 50%.

4. Мета дослідження полягає в розробці нейромережевої системи ідентифікації
особи для підприємства, що забезпечить автоматизацію контролю доступу, підвищення
рівня безпеки та мінімізацію впливу людського фактору на процес ідентифікації.

5. Зміст розрахунково-пояснювальної записки (основні задачі дослідження)

Огляд штучних нейронних мереж для ідентифікації особи;

Програмно-апаратні засоби нейромережевої системи ідентифікації особи на підприємстві;
Створення прототипу та експериментальні дослідження системи ідентифікації особи на підприємстві;
Аналіз результатів роботи розробленої нейромережевої системи ідентифікації;
Охорона праці та навколишнього середовища.

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Актуальність проекту;
Розробка архітектури нейромережевої системи ідентифікації особи;
Вибір програмно-апаратних засобів та мови програмування для створення системи;
Розробка нейромережевої системи для розпізнавання облич;
Реалізація модуля розпізнавання осіб;
Налаштування та тестування створеної системи;
Аналіз точності, продуктивності та впливових факторів розпізнавання системи;
Рекомендації щодо покращення точності та надійності системи.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Охорона праці	Герасін О.С., доцент каф. КСУ	25.11.2024 	05.12.2024 

Дата видачі завдання: « 02 » 09 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд штучних нейронних мереж для ідентифікації особи	15.10.2024	виконав
2	Програмно-апаратні засоби нейромережевої системи ідентифікації особи на підприємстві	04.11.2024	виконав
3	Створення прототипу та експериментальні дослідження системи ідентифікації особи на підприємстві;	18.11.2024	виконав
4	Аналіз результатів роботи розробленої нейромережевої системи ідентифікації	25.11.2024	виконав
5	Охорона праці та навколишнього середовища	05.12.2024	виконав
6	Оформлення пояснювальної записки	11.12.2024	виконав

Студент



(підпис)

Проворний І.В.

(прізвище та ініціали)

Керівник роботи



(підпис)

Герасін О.С.

(прізвище та ініціали)

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробці нейромережевої системи для ідентифікації особи на підприємстві, що забезпечує автоматизацію контролю доступу та підвищення рівня безпеки.

Робота має послідовну структуру. У першому розділі розглядаються сучасні методи розпізнавання облич, включаючи основи нейронних мереж, зокрема конволюційних, та способи їх навчання для ідентифікації осіб. У другому розділі розглядається вибір інструментальних засобів для реалізації системи, зокрема програмного забезпечення, бібліотек і середовища для роботи з нейронними мережами. У третьому розділі описується процес розробки системи ідентифікації особи, включаючи архітектуру моделі, формування бази даних ембедінгів та реалізацію алгоритмів розпізнавання осіб. У четвертому розділі проводиться аналіз роботи системи на основі тестування на реальних даних, оцінюється точність, продуктивність та надаються рекомендації для подальшого покращення.

Обсяг класифікаційної роботи становить 102 сторінки, 30 рисунків, 2 таблиці та 17 використаних джерел.

Ключові слова: нейромережева система, автоматизація контролю доступу, ідентифікація, модель.

ABSTRACT

This master's thesis is dedicated to the development of a neural network-based identification system for enterprises, aimed at automating access control and enhancing security levels.

The work is organized in a sequential structure. The first section reviews modern methods of face recognition, including the fundamentals of neural networks, particularly convolutional neural networks, and training techniques for identity verification. The second section discusses the selection of tools for implementing the system, including software, libraries, and environments for neural network development. The third section details the development process of the identification system, including the model architecture, embedding database formation, and implementation of face recognition algorithms. The fourth section analyzes the system's performance based on testing with real-world data, assessing accuracy, efficiency, and providing recommendations for further improvements.

The total length of the qualification paper is 102 pages, including 30 figures, 2 tables, and 17 references.

Keywords: neural network system, access control automation, identification, model.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ІДЕНТИФІКАЦІЇ ОСОБИ.....	10
1.1. Базові поняття нейронних мереж.....	10
1.2. Основні типи нейронних мереж.....	13
1.3. Аналіз існуючих систем для ідентифікації особи	17
1.4. Способи навчання нейромереж для вирішення завдань ідентифікації	20
Висновок до першого розділу.....	26
РОЗДІЛ 2. ПРОГРАМНО-АПАРАТНІ ЗАСОБИ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ НА ПІДПРИЄМСТВІ.....	27
2.1. Розробка архітектури системи.....	27
2.2. Вибір програмного середовища, мови програмування та бібліотек	28
Висновок до другого розділу	32
РОЗДІЛ 3. СТВОРЕННЯ ПРОТОТИПУ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ НА ПІДПРИЄМСТВІ	34
3.1. Розробка та інтеграція прототипу системи	34
3.2. Реалізація модуля розпізнавання осіб.....	55
3.3. Налаштування та тестування запропонованої системи	66
Висновок до третього розділу.....	71
РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ РОЗРОБЛЕНОЇ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ.....	73
4.1. Аналіз точності та продуктивності системи, впливальних на якість розпізнавання факторів.....	73
4.2. Рекомендації щодо покращення точності та надійності запропонованої системи	77
Висновок до четвертого розділу.....	78

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО	
СЕРЕДОВИЩА	80
5.1. Шкідливі та небезпечні фактори, які впливають на розробника інформаційної системи	81
5.2. Засоби зменшення впливу шкідливих та небезпечних факторів, які діють на розробника інформаційної системи	90
5.3. Охорона навколишнього природного середовища	95
Висновок до п'ятого розділу	99
ВИСНОВКИ	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	101

ВСТУП

У сучасному світі інформаційні технології набули широкого розповсюдження та стали важливим інструментом у багатьох сферах діяльності. Одним із значущих досягнень у цій галузі стало впровадження систем ідентифікації осіб, які знаходять застосування у різноманітних сферах, зокрема у забезпеченні безпеки підприємств. Ідентифікація осіб за допомогою нейромережових технологій забезпечує швидке і точне визначення особи, що сприяє покращенню рівня безпеки, ефективності роботи та автоматизації доступу на підприємстві.

Використання нейромереж для розпізнавання облич на підприємствах дозволяє вирішувати низку задач:

- система ідентифікації осіб забезпечує автоматизацію контролю доступу до приміщень. Завдяки ідентифікації працівників за їхнім обличчям, система дозволяє оперативно визначати осіб, які мають право входу, що сприяє зменшенню витрат на фізичну охорону;
- така система дозволяє підвищити рівень безпеки на підприємстві, знижуючи ймовірність несанкціонованого доступу до об'єктів. Завдяки високій точності нейронних мереж, система може швидко та ефективно розпізнавати обличчя навіть в умовах складного освітлення та різних кутів зйомки;
- нейромережові системи дозволяють створювати базу даних з ембедінгами облич працівників, що дає можливість швидко проводити аналіз для ідентифікації особи та відстеження переміщень на підприємстві.

У цій магістерській кваліфікаційній роботі буде розглянуто розробку нейромережової системи ідентифікації осіб на підприємстві, з урахуванням особливостей та переваг сучасних нейронних мереж. Результати дослідження та розробки дозволять створити надійну та ефективну систему, яка сприятиме підвищенню рівня безпеки та автоматизації підприємства.

Для досягнення поставленої мети в роботі передбачено виконання таких завдань:

1. Проаналізувати сучасні методи ідентифікації осіб за допомогою нейронних мереж;
2. Розробити архітектуру нейромережевої системи розпізнавання облич;
3. Реалізувати систему ідентифікації осіб з використанням конволюційної (CNN) нейронної мережі;
4. Провести тестування системи ідентифікації на реальних даних для оцінки її ефективності.

РОЗДІЛ 1

ОГЛЯД ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ІДЕНТИФІКАЦІЇ ОСОБИ

1.1. Базові поняття нейронних мереж.

Нейромережі, також відомі як штучні нейронні мережі (ШНМ), є комп'ютерними моделями, призначеними для моделювання роботи біологічних нейронних мереж людини. Вони використовуються для вирішення широкого спектра завдань, включаючи розпізнавання образів, аналіз даних, обробку природної мови, управління роботами та багато інших [4].

Нейромережі характеризуються наступними основними поняттями:

1. Нейрони. Основними будівельними блоками нейромережі є нейрони, які функціонально подібні до біологічних нейронів. Нейрони отримують вхідні сигнали, обробляють їх і передають результат наступному нейрону або шару нейронів;
2. Шари. Нейрони поєднуються в шари. Мережі можуть містити вхідний шар, приховані шари та вихідний шар. Вхідний шар отримує зовнішні дані, а вихідний шар видає результат роботи мережі;
3. Ваги та зв'язки. Кожен зв'язок між нейронами має вагу, яку визначає важливість цього зв'язку. Ваги налаштовуються у процесі навчання мережі;
4. Функції активації. Нейрони зазвичай реалізують функції активації до своїх входів, щоб ввести 10ік селів10сть у роботу мережі та дозволити їй моделювати складні залежності;
5. Навчання. Нейрони навчаються знаходити оптимальні ваги, які мінімізують помилку передбачень мережі. Навчання може бути наглядним (з учителем) чи ненаглядним (без учителя) залежно від завдання.

Біологічний аналог нейромереж – це нейронні мережі, що існують у людському мозку. Вони складаються з мільярдів нейронів, з'єднаних між собою через синапси. Робота біологічних нейронних мереж базується на передачі електричних імпульсів та хімічних сигналів між нейронами.

Важливі аспекти подібності між штучними нейромережами та біологічними нейронними мережами включають:

1. Обробка інформації. Як і біологічні нейрони, штучні нейрони обробляють інформацію, аналізуючи вхідні сигнали та передаючи їх далі;
2. Зв'язки та ваги. Аналогічно біологічним синапсам, у штучних нейромережах зв'язки між нейронами мають ваги, що регулюють передачу сигналу;
3. Навчання. Штучні нейромережі можуть навчатися, адаптуючись до даних, які і біологічні нейронні мережі, які змінюють свої зв'язки на основі досвіду;
4. Необхідність даних. Як біологічні нейрони вимагають вхідних сигналів до роботи, штучні нейромережі працюють з урахуванням вхідних даних.

Штучні нейромережі є математичними моделями, натхненими біологічними нейронами, і використовуються в багатьох додатках для аналізу даних та прийняття рішень.

Розглянемо принцип роботи нейромережі: на вхідний шар нейронів надходить певна інформація, яка по синапсах переходить до наступного шару. При цьому кожен синапс має свій власний коефіцієнт ваги, і будь-який наступний нейрон в новому шарі може мати кілька входів. Інформація передається далі, доки не досягне кінцевого виходу. Наприклад, алгоритм розпізнавання рукописного тексту повинен мати можливість справлятися з величезною різноманітністю способів представлення даних. Кожну цифру від 0 до 9 можна записати безліччю способів: розмір і точна форма кожного символу можуть відрізнятися залежно від того, хто пише і в яких обставинах.

Вхідному шару подаються значення, що становлять 12 пікселів, які становлять зображення рукописної цифри. Вихідний шар, у свою чергу, передбачає, який символ зображений на рисунку 1.1.

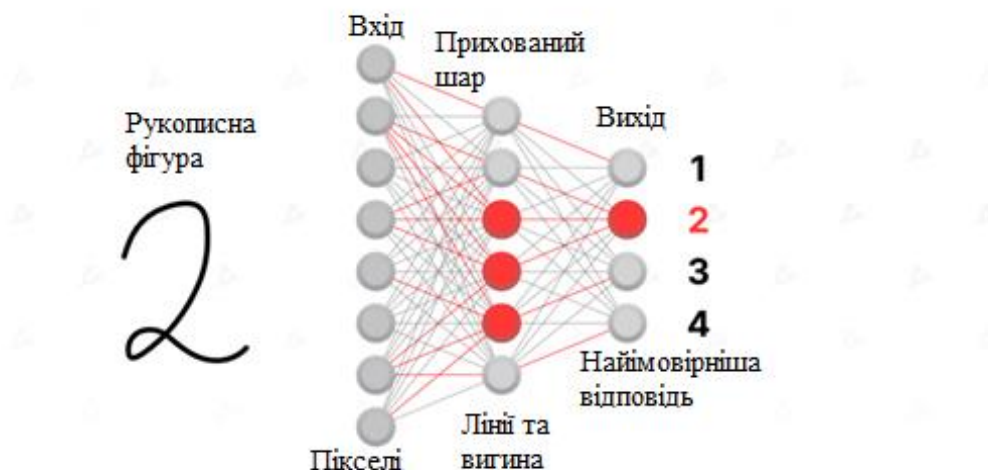


Рис.1.1. Принцип роботи неймережі

Коло на схемі являють собою нейрони, які організовані у зв'язані між собою вертикальні шари.

Кольори посилянь також різняться: вони позначають важливість зв'язків між нейронами. Червоні зв'язки посилюють значення при переході між шарами, що збільшує ймовірність активації нейронів у наступному шарі.

На схемі активовані нейрони заштриховані червоним. У “Прихованому шарі 1” вони означають, що зображення рукописної фігури містить певну комбінацію 12 пікселів, що нагадують горизонтальну лінію зверху рукописного числа 3 або 7.

Таким чином, “Прихований шар 1” може виявити характерні лінії та криві, які зрештою об'єднуються в повну рукописну фігуру.

Роль неймереж у сучасних технологіях:

- неймережі відіграють важливу роль у сучасних технологіях та багатьох галузях, включаючи комп'ютерний зір, обробку природної мови, медицину, робототехніку, фінанси та багато іншого;

- у галузі комп'ютерного зору нейромережі використовуються для розпізнавання образів, обробки зображено та відео, а також для вирішення завдань, пов'язаних із аналізом візуальних даних;
- в обробці природної мови нейромережі допомагають у завданнях автоматичного перекладу, генерації тексту, аналізі сентименту та багатьох інших задач, пов'язаних із текстом;
- у медицині нейромережі можуть використовуватися для діагностики та аналізу медичних зображень, а також для прогнозування захворювань та розробки ліків;
- у робототехніці нейромережі сприяють створенню автономних систем, здатних приймати рішення на основі даних із навколишнього середовища;

Огляд основних понять нейромереж та їх ролі у сучасних технологіях допомагає встановити фундаментальне розуміння теми перед глибшим розглядом різних аспектів нейромереж та їх практичним застосуванням у подальшій роботі.

1.2. Основні типи нейронних мереж.

Нейромережі є різноманітними моделями, які використовуються для обробки даних. Кожен тип нейромереж оптимальний для вирішення конкретних завдань і має характерні особливості. Далі доцільно докладніше розглянути кожен із них:

1) згорткові нейронні мережі (CNN) для обробки зображень. CNN – це Deep Learning-алгоритм, який може приймати вхідне зображення, присвоювати важливість (засвоювані ваги та зміщення) різним областям/об'єктам у зображенні і може відрізнити одне від одного (рис.1.2) [2];

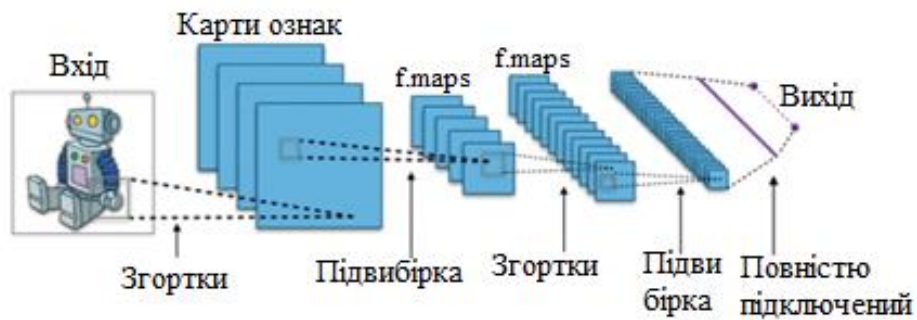


Рис.1.2. Згорткові нейронні мережі

2) глибокі нейронні мережі (DNN) для вивчення складних залежностей. DNN складаються з великої кількості шарів та використовуються для обробки складних ієрархічних даних. Вони дозволяють нейронним мережам вивчати ієрархічні уявлення даних на різних рівнях абстракції. Глибокі нейронні мережі застосовуються в багатьох областях, включаючи комп'ютерний зір, обробку природної мови та розпізнавання мови (рис.1.3) [5];

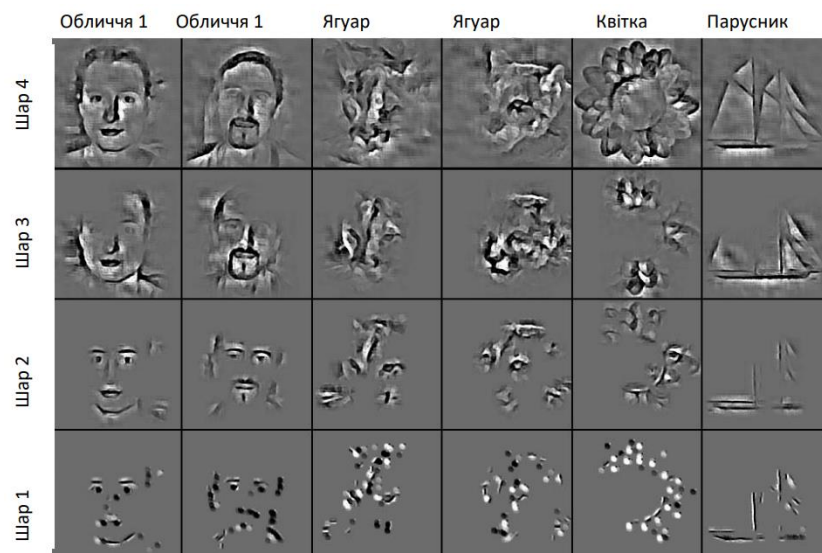


Рис.1.3. Результат обробки мережі DNN

3) мережі сіамських нейронів для порівняння зображень. Це спеціалізований тип нейронних мереж, який використовує однакові ваги при одночасній роботі з двома різними вхідними векторами для обчислення

вихідних векторів, що підлягають порівнянню. Дані мережі знаходять застосування в багатьох областях, включаючи біометричну ідентифікацію, пошук дублікатів фотографій, відслідковування об'єктів та багато інших завдань, де потрібно визначити схожість або відмінність між зображеннями (рис.1.4) [5];

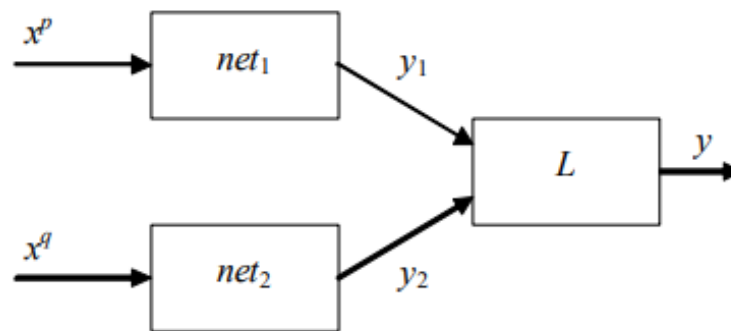


Рис.1.4. Схема сіамської нейронної мережі

4) рекурентні нейронні мережі (RNN) для послідовних даних. Використовуються для обробки даних, таких як часові ряди, тексти або мови. Вони мають спеціальні рекурентні зв'язки, які дозволяють враховувати контекст та залежності між послідовними елементами. RNN відрізняються тим, що потік інформації в них може рухатися у двох напрямках: нейрони отримують дані не лише від попереднього шару, а й від свого рекурентного зв'язку (рис.1.5) [5,16].

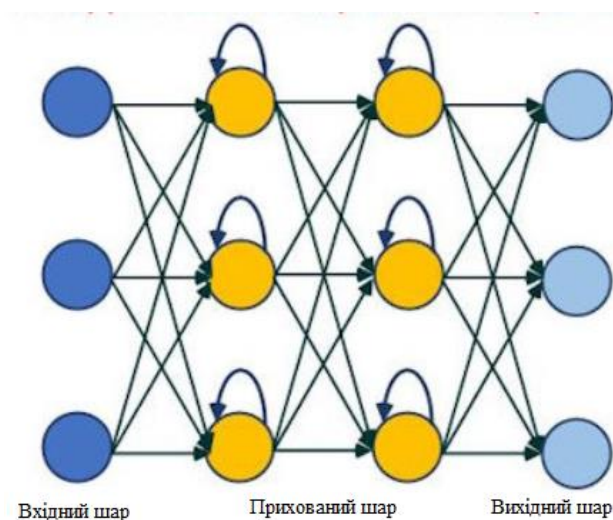


Рис.1.5. Рекурентна нейронна мережа

Кожен тип нейромереж має свої переваги та недоліки, і вибір певного типу залежить від завдання, доступних даних та ресурсів для навчання. Тому, доцільно провести порівняльний аналіз переваг і недоліків нейронних мереж для вибору відповідної, яка більше підходила для заданої теми.

Таблиця 1.1.

Порівняльна характеристика нейронних мереж

Тип нейромережі	Переваги	Недоліки
Згорткові нейронні мережі (CNN)	1) Ефективні для обробки зображень та відео; 2) Визначають важливі ознаки із візуальних даних	1) Вимагають великого обсягу даних для навчання; 2) Можуть бути обчислювально витратними
Глибокі нейронні мережі (DNN)	1) Моделюють складні нелінійні залежності; 2) Підходять для різнотипних завдань	1) Вимагають великого обсягу даних та обчислювальних ресурсів; 2) Проблема градієнта, що зникає
Мережі сіамських нейронів	1) Дозволяють порівнювати зображення; 2) Використовуються в біометричній ідентифікації	1) Потрібне навчання на парах зображень
Рекурентні нейронні мережі (RNN)	1) Обробляють послідовні дані; 2) Враховують контекст та залежності у даних	1) Проблема зникаючого градієнта; 2) Вимагають ретельного налаштування параметрів та архітектури

Ця таблиця надає огляд основних типів нейромереж. В даній роботі буде використано згорткові нейронні мережі (CNN), оскільки вони чудово підходять для завдання розпізнавання облич. CNN розроблені спеціально для обробки зображень та показали відмінну продуктивність у таких завданнях.

Також, для заданої задачі можна вибрати попередньо навчені моделі CNN, такі як VGG, ResNet або Inception, і до-навчити їх на власній задачі. Це може значно спростити процес навчання та покращити результати.

1.3. Аналіз існуючих систем для ідентифікації особи.

З кожним роком зростає інтерес до технологій розпізнавання облич, і зараз існує декілька десятків систем розпізнавання, які використовуються в різних сферах людського життя. Передовими компаніями, що розробляють такі системи, є: компанія NEC, Cognitec System GMBH, Neurotechnology.

1) Система *"FaceVACS-VideoScan"*. Система *"FaceVACS-VideoScan"* містить ряд системних компонентів, включаючи відео-сервер для керування відео-потокami, сервер відео-сканування, який координує всі частини системи та відповідає за основні біометричні операції, обчислювальний вузол для розподілу обчислювального навантаження, інтерфейс користувача, диспетчер сигналів для отримання повідомлень про події та обслуговуючий мобільні пристрої, операційну базу даних та набір інтеграторів [9].

Зараз технологія FaceVACS використовує алгоритм розпізнавання облич B10T9. Цей алгоритм володіє стійкістю до змін міміки, поворотів обличчя (на $\pm 15^\circ$), часткового закриття та використання окулярів, а також зміни освітлення.

Поміж особливостей системи FaceVACS можна виділити:

- можливість одночасного відстеження кількох облич;
- порівняння облич в реальному часі;
- здатність відображати та відправляти статистику про потоки;
- підтримка інтерактивної реєстрації з нерухомого пристрою;
- використання C++ API та Web Services API.

2) Система *"NEC's Face Recognition"* є однією з передових систем розпізнавання облич і була розроблена японською компанією NEC. Вона дозволяє ідентифікувати людей на основі кадрів, навіть якщо людина перебуває в окулярах або робить гримасу. Всі розпізнані особи зберігаються у базі даних, що дає можливість переглядати історію відео-реєстрації та визначати дату і час збереженого зображення [12].

Технологія NEC відзначається своєю високою точністю та швидкістю в порівнянні з безліччю інших систем розпізнавання. NEC аналізує індивідуальні особливості особи, такі як розмір, форма зіниць, лінії носа та рота, і використовує ці дані для пошуку в базі даних.

Система включає різні модулі, які реалізують різні алгоритми:

1. Метод узагальненої відповідності (GMFD), який забезпечує високу швидкість детектування та точність розпізнавання обличчя. Він базується на нейронних мережах та забезпечує попередній пошук пари очей;
2. Алгоритм PSM (Perturbation Space Method), який ефективно вирішує проблеми, пов'язані з розташуванням обличчя в кадрі, такі як обличчя під нахилом або під деяким кутом;
3. 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 обличч NeoFace має декілька особливостей:

- можливість спостереження та контроль в реальному часі;
- ідентифікація на основі індивідуальних особливостей особи;
- пошук подій в базі даних;
- стійкість до повороту особи на $\pm 15^\circ$ та нахилу голови 45° у будь-якому напрямку від фронтального положення;
- незалежне розпізнавання напрямку погляду та характеристик обличчя, наприклад, окуляри, борода або вираз обличчя.

3) Система *"LUNA SDK"* від компанії *"VisionLabs"* представляє собою спеціалізоване програмне забезпечення для супроводу та розпізнавання облич людей на цифрових фотографіях або відео-потоках. Цей двигун відзначається одними з найкращих у світі показників повноти та точності розпізнавання в реальних умовах [11].

Процес розпізнавання облич в LUNA SDK ґрунтується на глибоких нейронних мережах і складається з кількох ключових етапів: визначення положення та розмірів усіх облич (детекція), визначення розташування характерних рис особи та їх перетворення в стандартизовану форму (вирівнювання), вилучення дескрипторів (числових векторів, що узагальнюють характерні ознаки особи), порівняння облич з базою зображень та запобігання підміні облич.

Технічні характеристики та особливості LUNA SDK:

- здатність спостерігати та контролювати в реальному часі;
- підтримка різних платформ, включаючи Windows та Linux;
- повністю розроблено на C++;
- оптимізоване управління пам'яттю;
- використання алгоритму розпізнавання облич на основі глибоких нейронних мереж.

4) Система *"VeriLook SDK"* від компанії *"Neurotechnology"* є технологією ідентифікації облич, розробленою для виявлення та одночасного множинного розпізнавання людей на цифрових фотографіях та відео-потоках. Ця технологія відрізняється швидким і точним ідентифікуванням облич: вона здатна знаходити до 100 000 облич за секунду [10].

VeriLook SDK надається у формі комплекту для розробки програмного забезпечення і підтримує широкий спектр платформ, включаючи Windows, Linux, Mac OS X, iOS та Android.

Алгоритм VeriLook базується на глибоких нейронних мережах та реалізує локалізацію обличчя з використанням цифрових зображень.

Основні переваги системи VeriLook:

- одночасна обробка кількох облич;
- виявлення атрибутів особи, таких як посмішка, відкритий рот, закриті очі і т.д;
- “живе” розпізнавання обличчя: визначення, чи є особа у відеопотоці реальною або це лише фотографія;
- можливість порівнювати межі, захоплені в інфрачервоному спектрі;
- малий розмір шаблонів обличчя, розміром від 4 кілобайт.

Ці системи вирішують проблеми, пов'язані з ідентифікацією осіб, контролем доступу та безпекою на підприємствах та в інших контекстах. Вони допомагають підвищити ефективність і забезпечити високий рівень безпеки в різних сферах діяльності.

1.4. Способи навчання нейромереж для вирішення завдань ідентифікації.

Найважливішою властивістю нейронних мереж є їх здатність навчатися на основі даних про предметну область і в результаті навчання підвищувати свою продуктивність [20].

Процес навчання може розглядатися як визначення архітектури мережі і ітеративного налаштування вагових коефіцієнтів синаптичних зв'язків для ефективного виконання спеціальної задачі.

Під час навчання нейронної мережі передбачається наступна послідовність дій:

- нейронна мережа отримує вхідні сигнали із зовнішнього середовища;
- змінюються вільні параметри нейронної мережі, такі як ваги синапсів та зсуви (пороги) нейронів. Зміни параметрів можуть відбуватися під час навчання мережі, коли вона пристосовується до конкретної задачі;
- після зміни внутрішньої структури, нейронна мережа відповідає на вхідні сигнали іншим чином. Це може статися під час навчання, коли мережа навчається розпізнавати певні патерни або виконувати завдання. Визначення оптимальної структури мережі є окремою завданням, і воно залежить від конкретної задачі.

Для процесу навчання необхідно мати модель зовнішнього середовища, в якій функціонує нейронна мережа, тобто потрібну для вирішення задачі інформацію. По-друге, необхідно визначити, як модифікувати вагові параметри мережі. Алгоритм навчання означає процедуру, в якій використовуються правила навчання для налаштування ваг.

Модель дізнається, які зв'язки між нейронами є важливими для успішного прогнозування під час навчання. На кожному етапі тренування мережа використовує математичну функцію, щоб визначити, наскільки точним був останній прогноз порівняно з очікуваним.

Ця функція генерує серію значень помилок, які можуть використовуватися системою для розрахунку того, як модель повинна оновлювати значення ваги, прикріплених до кожного посилення, з кінцевою метою підвищення точності прогнозів мережі.

Протягом багатьох тренувальних циклів і за допомогою періодичного ручного налаштування параметрів мережа продовжуватиме генерувати все більш точні прогнози, доки не досягне максимального значення. На цьому етапі, наприклад, коли рукописні цифри можна розпізнати з точністю понад 95%, можна сказати, що нейромережа навчена.

Існує три загальні парадигми навчання [23]:

- контрольне навчання (з вчителем);
- не контрольне навчання (без вчителя, самонавчання);
- змішане навчання.

У першому випадку нейромережа має у своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді як можна більш близькі до відомих правильних відповідей.

Навчання без вчителя не вимагає знання правильних відповідей на кожен приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних та кореляція між зразками в навчальній множині, що дозволяє розподілити зразки по категоріях.

При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час як інша визначається за допомогою самонавчання.

Оцінка ефективності навчання нейромережі залежить від кількох керованих факторів. Теорія навчання розглядає три фундаментальні властивості, пов'язані з навчанням: ємність, складність зразків і обчислювальна складність:

- ємність вказує, скільки зразків може запам'ятати мережа, і які межі прийняття рішень можуть бути на ній сформовані;
- складність зразків визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення;
- обчислювальна складність напряму пов'язана з потужністю процесора ЕОМ.

Для нейронних мереж існує багато правил навчання, але більшість з цих правил є деякою зміною відомого та найстаршого правила навчання – правила Хебба [22]:

- правило Хебба: «якщо нейрон отримує вхідний сигнал від іншого нейрона і обидва є високо активними (математично мають такий самий знак), вага між нейронами повинна бути підсилена». При збудженні одночасно двох нейронів з виходами (x_j, y_i) на k -тому кроці навчання вага синаптичного з'єднання між ними зростає, в інакшому випадку – зменшується, тобто: $\Delta W_{ij}(k) = r * x_j(k) * y_i(k)$, де r – коефіцієнт швидкості навчання. Може застосовуватися при навчанні “з вчителем” і “без вчителя”;

- правило Хопфілда: є подібним до правила Хебба за винятком того, що воно визначає величину підсилення або послаблення. «Якщо одночасно вихідний та вхідний сигнал нейрона є активними або неактивними, збільшуємо вагу з'єднання оцінкою навчання, інакше зменшуємо вагу оцінкою навчання»;

- правило “дельта”: це правило є подальшою зміною правила Хебба і є одним із найбільш загально використовуваних. Це правило базується на простій ідеї неперервної зміни синаптичних ваг для зменшення різниці ("дельта") між значенням бажаного та біжучого вихідного сигналу нейрона: $\Delta W_{ij} = x_j(d_i - y_i)$. У правилі «дельта» похибка, що отримана у вихідному прошарку перетворюється похідною передатної функції і послідовно пошарово поширюється назад на попередні прошарки для корекції синаптичних ваг. При використанні правила «дельта» важливим є невпорядкованість множини вхідних даних;

- правило градієнтного спуску: це правило подібне до правила «дельта» використанням похідної від передатної функції для змінювання похибки «дельта» перед тим, як застосувати її до ваг з'єднань. До кінцевого коефіцієнта зміни, що діє на вагу, додається пропорційна константа, яка пов'язана з оцінкою навчання. І хоча процес навчання збігається до точки

стабільності дуже повільно, це правило поширене і є загально використовуване;

- правило змагання: на відміну від правила Хебба, в якому множина вихідних нейронів може збуджуватись одночасно, при навчанні методом змагання вихідні нейрони змагаються між собою за активізацію. Це явище відоме як правило "переможець отримує все". Подібне навчання має місце в біологічних нейронних мережах. Навчання за допомогою змагання дозволяє кластеризувати вхідні дані: подібні приклади групуються мережею відповідно до кореляцій і представляються одним елементом:

$$\Delta W_{ij}(k + 1) = W_{ij}(k) + r[x_j - W_{ij}(k)].$$

Також, невід'ємну роль у навчанні нейромережі грають дані та бази даних. Роль даних у навчанні нейромереж є критичною, оскільки якість і кількість даних мають прямий вплив на продуктивність та узагальнюючу здатність моделі. Ось ключові аспекти ролі даних:

1. Збір даних. Збір даних – перший та найважливіший крок. Дані можуть бути зібрані з різних джерел, включаючи зображення, текст, звук, відео та інші. Ефективне складання даних визначає, наскільки добре модель зможе вирішувати завдання;

2. Підготовка даних. Дані часто потребують попередньої обробки, такої як масштабування, нормалізація, очищення та кодування. Цей процес забезпечує однаковість даних та робить їх придатними для навчання;

3. Аугментація даних. Аугментація даних – це метод розширення набору даних шляхом створення варіацій існуючих даних. Наприклад, для зображень можна повертати, змінювати розмір та змінювати яскравість, щоб збільшити різноманітність даних та покращити узагальнення моделі;

4. Навчальний, валідаційний та тестовий набори даних. Дані зазвичай поділяються на три частини: навчальний набір для навчання моделі, валідаційний набір для налаштування гіперпараметрів та оцінки

продуктивності, та тестовий набір для остаточної перевірки моделі. Це допомагає уникнути перенавчання;

5. Бази даних. Великі обсяги даних часто зберігаються у спеціалізованих базах даних. Бази даних надають механізми для зберігання, організації та швидкого доступу до даних.

Також, зворотне поширення помилок (backpropagation) є ключовим методом навчання нейронних мереж. Цей процес включає обчислення градієнта функції втрат по відношенню до ваг мережі і використання цього градієнта для оновлення ваг таким чином, щоб мінімізувати функцію втрат. Далі цей процес розглядається детальніше:

1. Пряме поширення. На початку дані пропускаються через нейронну мережу в прямому напрямку. Кожен нейрон виконує лінійне перетворення вхідних даних та застосовує функцію активації до результату. Процес прямого поширення призводить до отримання передбачень мережі;

2. Обчислення помилки. Після отримання передбачень обчислюється помилка шляхом порівняння передбачених значень із фактичними даними. Це відбувається з використанням функції втрат, яка вимірює різницю між прогнозами та даними;

3. Зворотне поширення помилки. У цьому кроці обчислюється градієнт функції втрат по відношенню до кожної ваги мережі з використанням правила ланцюга. Цей градієнт є мірою, як вага повинна бути змінена, щоб зменшити помилку;

4. Оновлення ваг. Градієнти, обчислені в попередньому кроці, використовуються для оновлення ваги мережі за допомогою методу оптимізації, такого як стохастичний градієнтний спуск (SGD). Вага оновлюється так, щоб мінімізувати помилку, і процес зворотного поширення помилки повторюється на безліч епох;

5. Збіжність. Процес навчання триває доти, доки функція втрат не зійдеться до мінімуму, і модель буде готова до використання нових даних.

Базові поняття, які були розглянуті в цьому розділі, відіграють важливу роль в успішному навчанні нейромереж, і розуміння їх допомагає розробникам та дослідникам створювати ефективні та надійні моделі машинного навчання. Важливо продовжувати вивчати та застосовувати ці концепції для досягнення найкращих результатів у галузі глибокого навчання.

Висновок до першого розділу.

У першому розділі проведено детальний огляд принципів роботи штучних нейронних мереж, зокрема їхнього використання для розв'язання задач розпізнавання облич.

Розглянуто різні типи нейронних мереж, включаючи згорткові нейронні мережі (CNN), які виявилися найбільш ефективними для ідентифікації облич через їхню здатність аналізувати просторові ієрархії в зображеннях.

Описано основні етапи навчання нейронних мереж, такі як навчання з учителем та без учителя. Крім того, проаналізовано існуючі системи розпізнавання облич та методи їх навчання.

Для подальшої реалізації системи розпізнавання осіб вибрано CNN-мережі через їхню високу точність та ефективність при роботі із зображеннями.

РОЗДІЛ 2

ПРОГРАМНО-АПАРАТНІ ЗАСОБИ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ НА ПІДПРИЄМСТВІ

2.1. Розробка архітектури.

Для того щоб пояснити архітектуру нейромережевої системи ідентифікації особи на підприємстві, варто розпочати з огляду основних складових і принципів роботи такої системи. Система ідентифікації облич базується на інтеграції декількох компонентів, кожен з яких виконує свою ключову функцію. Це дозволить забезпечити взаємодію між користувачем, базою даних і алгоритмом розпізнавання в реальному часі.

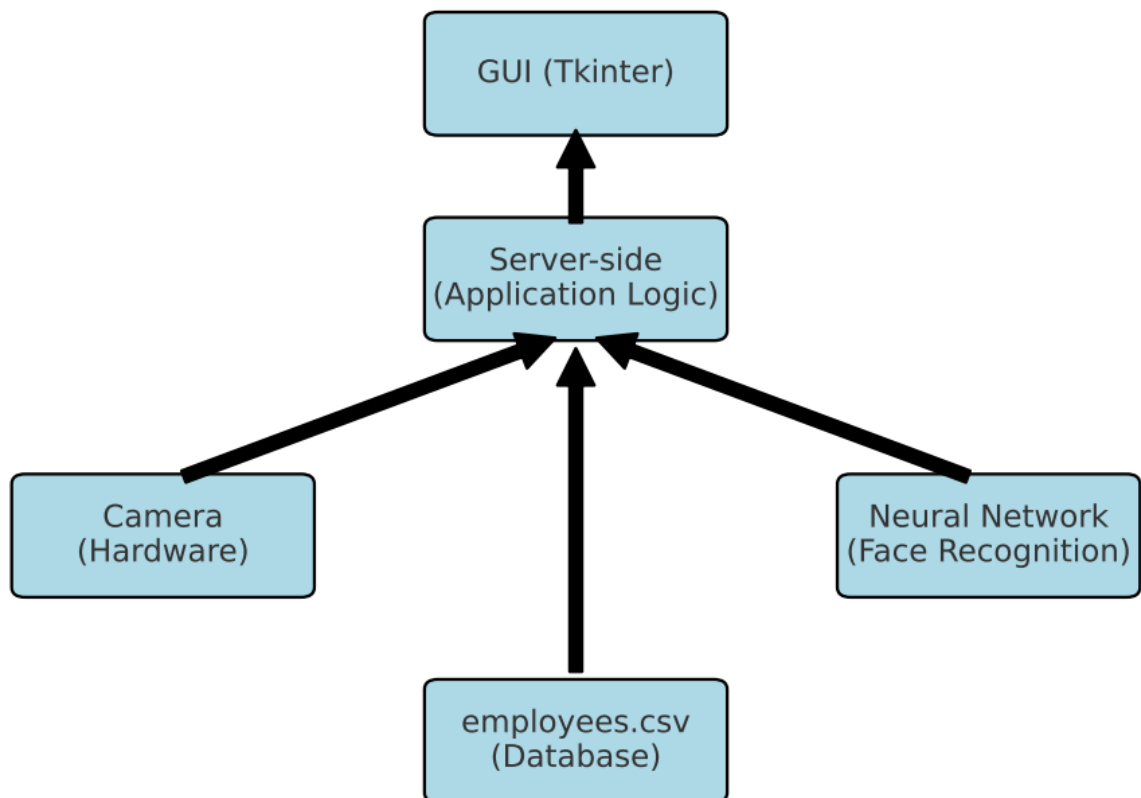


Рис.2.1. Елементи архітектури по розпізнаванню облич

Основними елементами архітектури є:

1. Клієнтська частина (GUI). Забезпечує взаємодію користувача із системою, включаючи інтерфейс для запуску процесу ідентифікації, відображення результатів, а також функції для збереження зображень. Це

графічний інтерфейс, побудований на основі бібліотеки Tkinter, який дозволяє користувачу виконувати основні операції системи;

2. Сторона сервера (Server Side). Відповідає за роботу з базою даних співробітників (надалі для цього буде створено файл `employees_inform.csv`) і процес розпізнавання облич. Сервер обробляє запити з клієнтської частини та надає необхідні дані з бази або результати розпізнавання;

3. Модуль нейронної мережі (Neural Network). Центральний компонент системи, що відповідає за обробку зображень, отриманих з камери, і порівнянням їх із зображенням у базі даних;

4. База даних (Database). У нашій системі буде використовуватися файл `employees_inform.csv`, який містить дані про співробітників (ID, ім'я, прізвище, посада, шлях до зображення);

5. Взаємодія з обличчям (Camera). Камера захоплює зображення облич користувачів, яке потім передається на обробку в модуль розпізнавання. Це апаратний компонент, інтегрований із системою для забезпечення ідентифікації в реальному часі.

Використана архітектура є модульною, де кожен компонент виконує чітко визначену функцію. Це дозволяє забезпечити легкість у підтримці та масштабованість системи. Поділ на клієнтську та серверну частину сприяє підвищенню продуктивності, а використання нейронних мереж підвищує точність ідентифікації.

2.2. Вибір програмного середовища, мови програмування та бібліотек

1) Для розробки системи ідентифікації буде використано таке програмне забезпечення, як: Visual Studio Code.

Visual Studi Code – це спрощений редактор коду з підтримкою таких операції розробки, як налагодження, виконання завдань і керування версіями.

В даному редакторі коду є можливість підключення різних плагінів, що дає можливість працювати з різними мовами програмування для розробки будь-якого ІТ-продукту.

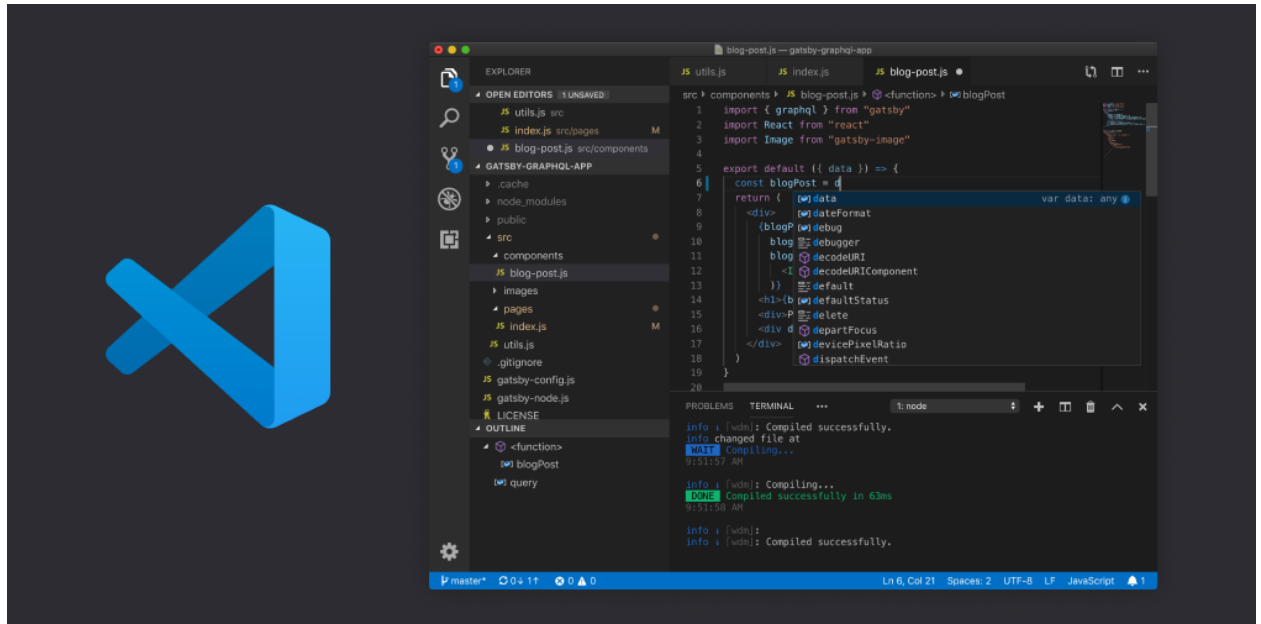


Рис.2.2. Visual Studio Code

2) Мову програмування, яка буде використана для проекту – це Python. Python є одним із найпопулярніших мов програмування для створення нейронних мереж та систем розпізнавання облич. Це пояснюється кількома важливими перевагами:

- простота у використанні. Python має простий синтаксис, що робить його легким для розуміння і навчання. У проектах на основі нейронних мереж це дозволяє зосередитися на логіці побудови моделей і підготовці даних;
- потужні бібліотеки для машинного навчання. Ця мова має велику кількість бібліотек для роботи з нейронними мережами та обробки даних, зокрема такі як: TensorFlow, PyTorch, Keras, NumPy, Pandas. Ці бібліотеки містять оптимізовані алгоритми для роботи з великими масивами даних і дозволяють швидко створювати та тренувати моделі нейронних мереж;

- інтеграція з науковими інструментами. Python широко використовується в наукових дослідженнях завдяки бібліотекам для статистичного аналізу, обробки зображень (Pillow), і машинного навчання;
- кросплатформеність. Python є кросплатформеним, що дозволяє запускати додатки на різних операційних системах (Windows, Linux, macOS), що дуже важливо для корпоративних рішень. Це забезпечує гнучкість та легкість впровадження на різних підприємствах.

Python підходить для розробки нейромережових систем завдяки своїй універсальності, потужним інструментам для обробки даних, підтримці нейронних мереж та інтенсивній спільноті, що забезпечує постійний розвиток бібліотек і алгоритмів для вирішення завдань розпізнавання облич.

3) У проекті нейромережової системи ідентифікації облич використання зовнішніх бібліотек є ключовим фактором, що значно прискорює розробку та забезпечує ефективність. Нижче описані основні бібліотеки, які будуть використані при виконанні проекту:

1. Tkinter – це стандартний інтерфейс Python до інструментарію GUI. Ця бібліотека використовується для побудови графічного інтерфейсу користувача;
2. OpenCV – використовується для обробки зображень, захоплення облич з камери та попередньої підготовки даних для нейромережі;
3. TensorFlow / Keras – спрощена конструкція нейронної мережі. Використовуються для побудови та навчання моделі нейронної мережі;
4. Pandas – бібліотека для роботи з даними у форматі CSV. Буде використовуватися для читання та обробки файлів CSV у проекті;
5. PIL (Pillow) – використовується для обробки зображень у Python. Вона надає можливість відкривати, обробляти, змінювати розміри та зберігати зображення у різних форматах;
6. Torch (PyTorch) – одна з найпопулярніших бібліотек для створення та тренування нейронних мереж;

7. NumPy – це фундаментальна бібліотека для наукових обчислень на Python, яка забезпечує швидку роботу з багатовимірними масивами та матрицями даних.

Інтеграція цих бібліотек дозволить системі ідентифікації облич працювати швидко, ефективно та надійно.

4) Віртуальне середовище (venv). Для забезпечення ізоляції залежностей у проекті була використана технологія venv (див. Табл. 2.1). Це дозволило створити окреме середовище для розробки, в якому всі зовнішні бібліотеки і пакети були встановлені локально для цього проекту, не впливаючи на інші проекти та системні бібліотеки.

Використання venv дає можливість уникнути конфліктів версій пакетів та залежностей, які можуть виникати при роботі над кількома проектами на одній машині. Це особливо важливо, коли проект потребує специфічних версій бібліотек, що можуть відрізнятися від тих, які використовуються в інших проектах.

Завдяки цьому, весь процес розробки та тестування програми відбувається в ізольованому та контрольованому середовищі, що сприяло стабільній та надійній роботі системи. Крім того, ізоляція середовища дозволила значно спростити процес розгортання проекту на нових машинах: копіювання та відновлення залежностей у venv займає мінімум часу, а встановлені бібліотеки відповідають точно таким самим версіям, які були використані при початковій розробці.

Таблиця 2.1.

Порівняльна таблиця використання venv

Критерії	З використанням venv	Без venv
Ізоляція залежностей	Кожен проект має окреме середовище з власними	Всі проекти використовують глобальні залежності, що може

	залежностями.	призвести до конфліктів версій.
Конфлікти між проектами	Виключаються завдяки окремим середовищам для кожного проекту.	Можливі конфлікти версій бібліотек між проектами.
Легкість відтворення середовища	Легке відтворення за допомогою requirements.txt або аналогічних файлів.	Відтворення середовища ускладнене через можливі конфлікти та глобальні залежності.
Керування бібліотеками	Просте встановлення та видалення бібліотек лише в контексті проекту.	Встановлення/видалення бібліотек впливає на всі проекти у системі.
Стабільність проекту	Забезпечує стабільність, оскільки зміни в інших проектах не впливають на поточний.	Можливі проблеми через зміни в глобальних бібліотеках, що використовуються іншими проектами.
Встановлення бібліотек	Встановлюються локально всередині віртуального середовища.	Встановлюються глобально, впливаючи на всі проекти в системі.
Швидкість тестування та розробки	Немає потреби перевіряти конфлікти бібліотек, що прискорює тестування та розробку.	Час витрачається на виправлення конфліктів залежностей.
Залежність від операційної системи	Менше залежить від глобальних налаштувань ОС.	Глобальні залежності ОС можуть впливати на роботу проекту.
Чистота системи	Система залишається чистою, оскільки залежності зберігаються локально для кожного проекту.	Глобальні бібліотеки можуть накопичуватися і впливати на інші проекти.

Висновок до другого розділу

У даному розділі детально розглянуто програмно-апаратні засоби нейромережевої системи ідентифікації особи на підприємстві. Спочатку

розроблено архітектуру системи, що включає клієнтську частину для взаємодії з користувачем, серверну частину для обробки запитів і базу даних, що містить інформацію про співробітників. Це забезпечило модульність системи та легкість у подальшій підтримці.

Мова програмування Python та редактор коду Visual Studio Code вибрані для подальшого створення системи, оскільки ці інструменти надають зручний інтерфейс для розробки та підтримують потужні бібліотеки для роботи з нейронними мережами. Також використання віртуального середовища (venv) сприяє ізоляції залежностей і стабільній роботі системи.

У рамках даного розділу закладено основу для подальшої розробки та впровадження системи, що дозволить ефективно здійснювати ідентифікацію осіб на підприємстві, що є важливим аспектом для забезпечення безпеки та оптимізації бізнес-процесів.

РОЗДІЛ 3

СТВОРЕННЯ ПРОТОТИПУ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОСОБИ НА ПІДПРИЄМСТВІ

3.1. Розробка та інтеграція прототипу системи

Процес розробки та інтеграції системи ідентифікації буде включати кілька ключових етапів: навчання моделі, створення архітектури нейронної мережі, налаштування параметрів та подальше тестування. На першому етапі здійснюється підготовка даних та початкове навчання моделі з використанням спеціально розробленого скрипту `train.py`. Далі відбувається побудова архітектури моделі в файлі `YourModel.py`, яка відповідає за обробку та класифікацію зображень об'єктів. Заключний етап включає перевірку та налаштування основних параметрів, щоб забезпечити високу точність та ефективність системи ідентифікації.

Важливими складовими системи є також набори даних, які використовуються для навчання моделі та тестування її продуктивності. У цьому розділі буде представлено детальний огляд кожного з етапів, включаючи навчання моделі, структуру нейронної мережі та особливості роботи з набором даних.

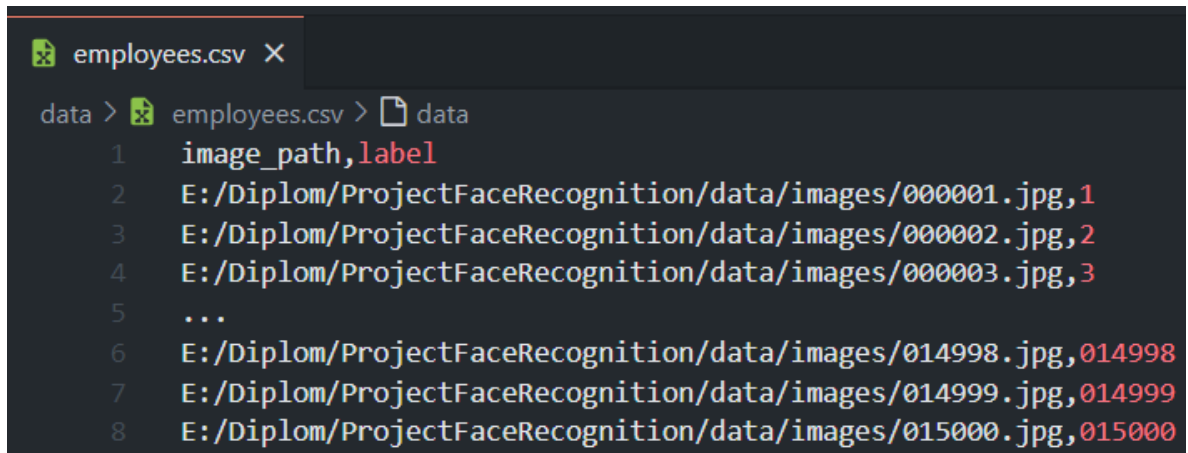
Перед тим як перейти до опису процесу навчання моделі, важливо розібратися, як використовується набір даних для цієї мети. У рамках проекту створюється папка `data`, всередині якої розміщуються файл `employees.csv` та папка `images`:

- 1) `images` – в цій папці розміщений великий набір фотографій, який використовуватиметься на різних етапах навчання моделі;
- 2) `employees.csv` – цей файл слугує основою для навчання моделі. Він містить шляхи до зображень, що знаходяться в папці `images`, а також

відповідні мітки, які ідентифікують кожне зображення. Цей набір даних, надалі, буде підключатися до файлу `train.py` за допомогою шляху:

```
csv_file_path = "E:/Diplom/ProjectFaceRecognition/data/employees.csv"
```

Цей файл є дуже важливим для навчання, адже він містить необхідні дані для прив'язки кожного зображення до певного ідентифікатора, що дозволяє алгоритму навчатися зіставляти обличчя з відповідними мітками.



```
employees.csv X
data > employees.csv > data
1 image_path, label
2 E:/Diplom/ProjectFaceRecognition/data/images/000001.jpg, 1
3 E:/Diplom/ProjectFaceRecognition/data/images/000002.jpg, 2
4 E:/Diplom/ProjectFaceRecognition/data/images/000003.jpg, 3
5 ...
6 E:/Diplom/ProjectFaceRecognition/data/images/014998.jpg, 014998
7 E:/Diplom/ProjectFaceRecognition/data/images/014999.jpg, 014999
8 E:/Diplom/ProjectFaceRecognition/data/images/015000.jpg, 015000
```

Рис.3.1. Приклад навчання на наборі з 15 000 даних

Далі, у проєкті, створюємо папку `src`, де будуть знаходитися файли для навчання, створення моделі та у майбутньому файл для запуску всього проєкту.

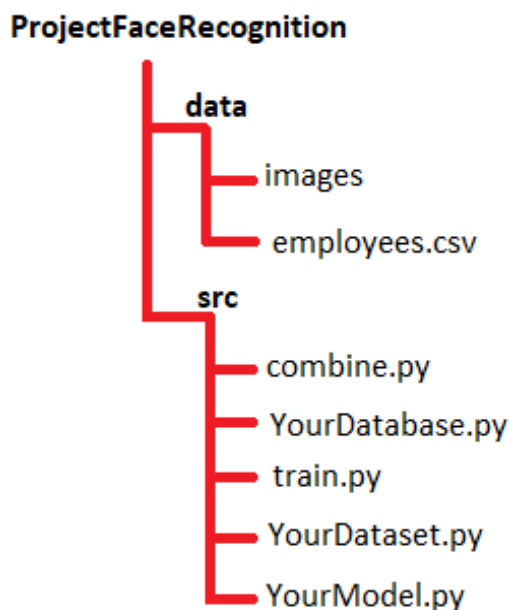


Рис.3.2. Структура проєкту

Розробивши структуру нашого набору даних та структуру самого проекту, можемо перейти до процесу навчання моделі. Навчання моделі – є важливим етапом, на якому система ідентифікації осіб буде формуватися за допомогою підготовлених даних.

Процес навчання моделі включає кілька ключових етапів, які будуть змінювати свій зміст протягом кількох стадій навчання:

- підготовка даних – на цьому етапі імплементуємо код у файлі `YourDataset.py`, що відповідає за завантаження та обробку наших зображень та міток. Це необхідно для того, щоб модель могла навчатися на чистих і структурованих даних;
- архітектура моделі – в файлі `YourModel.py` визначаємо архітектуру нашої нейронної мережі. Це важливо, оскільки архітектура безпосередньо впливає на здатність моделі навчатися і здійснювати точні прогнози;
- навчання моделі – сам процес навчання реалізовано у файлі `train.py`. В даному файлі налаштовуємо параметри моделі, використовуємо різні техніки оптимізації та виконання тренувань на нашому наборі даних.

Для детального розуміння процесу роботи нейромережевої системи ідентифікації особи, яку обрано в якості основи нейромережевої архітектури, що базується на згортковій нейронній мережі (CNN), розглянемо структурну схему нейронної мережі, що застосовується для аналізу зображень облич (рис.3.3). Дана схема відображає кінцеву архітектуру нейромережевої системи, яка реалізується у файлі `YourModel.py`. Ця схема має основні шари мережі та їхні функції, починаючи з вхідного шару, де отримується зображення, і до вихідного шару, який видає результат ідентифікації. На кожному етапі обробки зображень відбувається поступове виділення ознак, що дозволяє системі ефективно розпізнавати обличчя навіть при складних умовах.

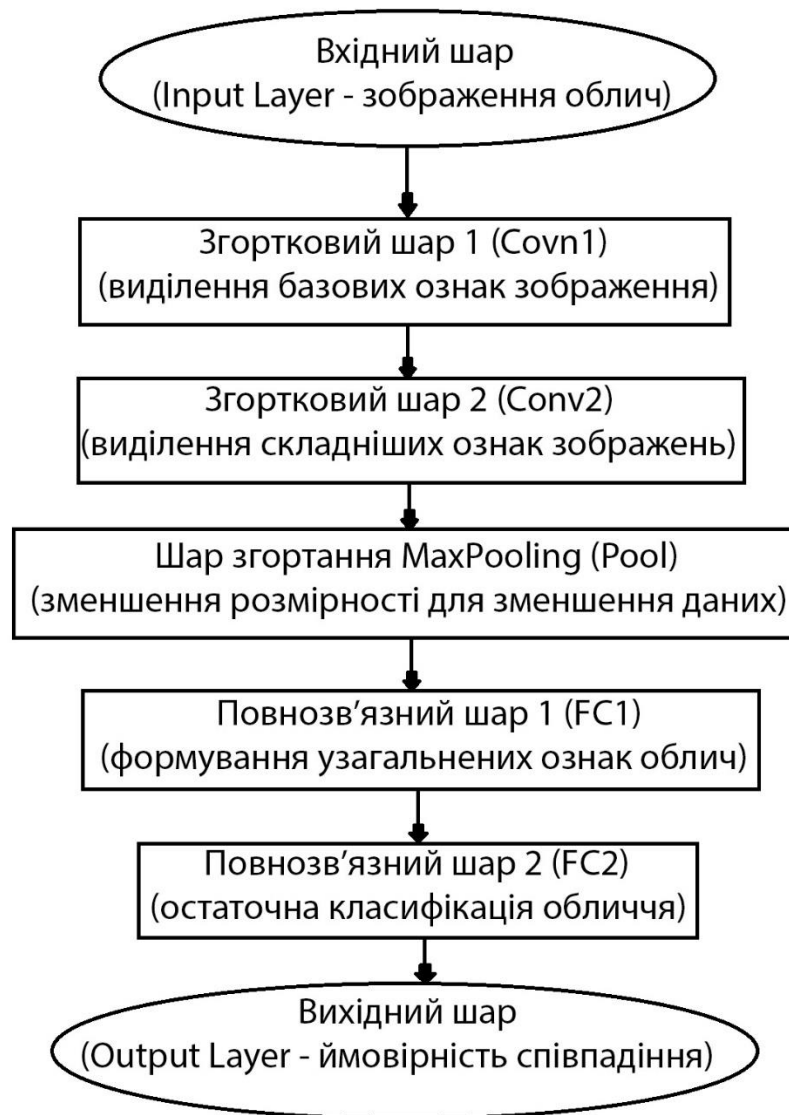


Рис.3.3. Кінцева архітектура нейромережевої системи

Опис алгоритму роботи нейронної мережі для ідентифікації:

1. Вхідний шар (Input Layer): на вхід подається зображення обличчя, яке переводиться у числовий формат (пікселі). Цей шар приймає дані та передає їх на наступний етап обробки;
2. Перший згортковий шар (Conv1): на даному етапі відбувається виділення базових ознак зображення, таких як краї та текстури. Цей шар обробляє зображення за допомогою фільтрів і передає дані на наступний згортковий шар;
3. Другий згортковий шар (Conv2): тут виділяються складніші ознаки зображення, такі як форми і шаблони. Це допомагає мережі краще

розпізнавати особливості обличчя, поступово переходячи від базових до більш детальних ознак;

4. Шар згортання (MaxPooling Layer): після згорткових шарів дані проходять через шар MaxPooling, який зменшує розмірність ознак (це зменшує обсяг даних та обчислювальні витрати, зберігаючи важливі особливості);

5. Перший повнозв'язний шар (FC1): на цьому етапі відбувається об'єднання всіх виділених ознак для формування більш узагальнених характеристик обличчя. Цей шар готує дані для остаточної класифікації;

6. Другий повнозв'язний шар (FC2): цей шар виконує остаточну класифікацію, аналізуючи узагальнені ознаки та визначаючи належність до певного класу;

7. Вихідний шар (Output Layer): на виході мережа видає ймовірність збігу, що вказує на ступінь відповідності введеного зображення одному з відомих облич. Ця ймовірність інтерпретується як результат ідентифікації.

Цей алгоритм відображає поетапну обробку зображень, починаючи з виділення базових ознак і закінчуючи остаточною класифікацією для ідентифікації обличчя.

Тепер можна переходити до детального розгляду кожного з етапів. Також спробуємо відобразити за допомогою структурних схем процеси, які відбуваються на кожному з етапів навчання:

1. Перше етап навчання: цей етап зосереджується на підготовці даних та навчанні моделі. У цьому етапі закладаємо основу для подальшої роботи, де відбувається не лише навчання моделі, а й формування її архітектури.

- Файл YourDataset.py: цей файл відповідає за створення датасету для навчання моделі. Його структурна схема зображена на рис.3.4

Клас YourDataset успадковує базовий клас Dataset із PyTorch

У конструкторі `__init__` завантажується CSV файл з анотаціями (де вказані шляхи до зображень та їх мітки), а також задається коренева папка, в

якій зберігаються зображення. Перетворення, такі як нормалізація або зміна розміру, можуть бути передані через параметр `transform`

Метод `__len__` повертає кількість елементів у датасеті, що потрібне для ітерації за даними

Метод `__getitem__` завантажує конкретне зображення та його мітку за індексом. Зображення відкривається і при необхідності до нього використовуються перетворення. Далі повертається пара “зображення – мітка”, яка використовується в процесі навчання моделі

Цей код дозволяє ефективно завантажувати та обробляти дані для навчання нейронної мережі;

```
YourDataset.py, first training  
  
import os, pandas, Dataset, transforms, Image  
  
YourDataset(Dataset)  
    __init__(self, csv_file, root_dir, transform)  
        self.annotations = pd.read_csv(csv_file)  
        self.root_dir = root_dir  
        self.transform = transform  
  
    __len__(self)  
        return len(self.annotations)  
  
    __getitem__(self, idx)  
        img_path = os.path.join(..)  
        image = Image.open(..).convert("RGB")  
        label = int(self.annotations.iloc[idx, 1])  
        if self.transform  
            image = self.transform(image)  
        return image, label
```

Рис.3.4. Структурна схема файлу `YourDataset, first training`

- Файл `YourModel.py`: цей файл відповідає за визначення архітектури нейронної мережі. Його структурна схема зображена на рис.3.5

`YourModel` успадковує клас `nn.Module` з `PyTorch`, що дозволяє йому бути сумісним з його механізмами.

У методі `__init__` задаються шари мережі: два згорткових шари (Conv2D), які обробляють зображення, і два повнозв'язних шари (Linear), які перетворюють вихід згорткових шарів на остаточні передбачення.

Метод `forward` описує, як дані проходять через шари моделі. Вхідні дані (в нашому випадку – це зображення) спочатку проходять через згорткові шари з функціями активації ReLU та операціями MaxPool для зменшення розмірності. Після цього дані перетворюються на одновимірний масив (flatten), а потім через два повнозв'язкові шари до фінального виходу – ймовірностей класів.

Таким чином, цей код описує базову згорткову нейронну мережу (CNN), яка може вирішувати завдання класифікації зображень;

YourModel.py, first training

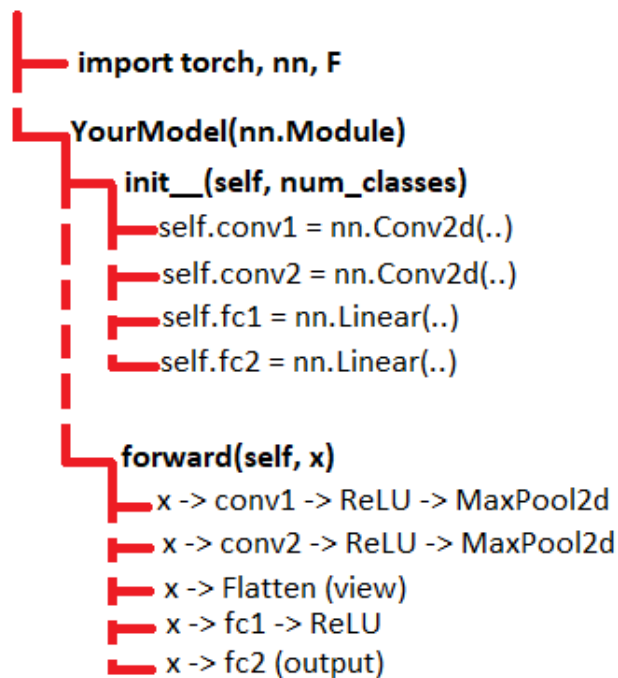


Рис.3.5. Структурна схема файлу `YourModel`, first training

- Файл `train.py`: основна функція цього файлу – навчання нейронної мережі. Його структурна схема зображена на рис.3.6

На початку створюються та ініціалізуються всі необхідні елементи: імпортуються бібліотеки, задаються гіперпараметри (розмір батча, швидкість навчання, кількість епох та класів).


```
import torch, optim, F, DataLoader, YourDataset, YourModel, transforms
device = torch.device(..)
batch_size, learning_rate, epochs, num_classes
csv_file_path, images_folder_path, model_weight_path

Dataset creation
transform = transforms.Compose([..])
your_dataset = YourDataset(..)

if dataset is not empty
    DataLoader = DataLoader(your_dataset,..)
    model = YourModel(..).to(device)
    optimizer = optim.Adam(..)

    Training loop (for each epoch)
        model.train()
        total_correct, total_samples
        For each batch in dataloader
            images, labels = batch
            images, labels -> to(device)
            optimizer.zero_grad()
            outputs = model(images)
            loss = F.cross_entropy(outputs, labels)
            loss.backward()
            optimizer.step()
            Calculate accuracy
        Print epoch, loss, accuracy
    Save model weights (torch.save(..))
```

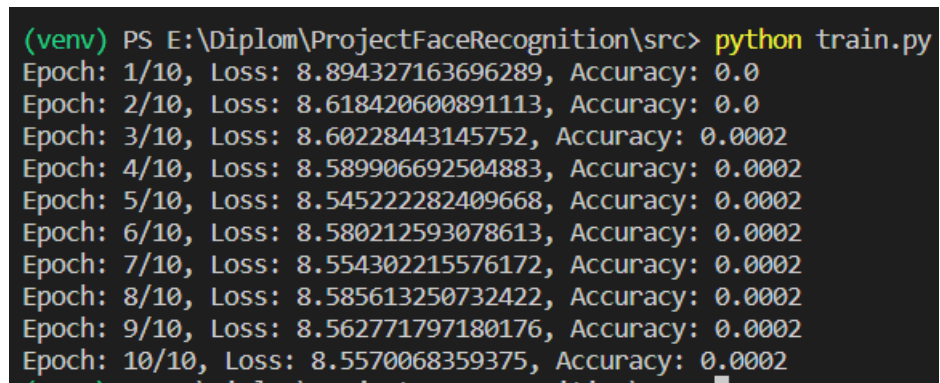
У першому навчанні, вони мали такі параметри: `batch_size = 32`; `learning_rate = 0.005`; `epochs = 10`; `num_classes = 5001`. Далі створюється датасет із застосуванням перетворень, і цей датасет використовується в `DataLoader` для ітерації за даними.

Потім створюється модель та оптимізатор для її навчання. Основна частина коду – це цикл навчання, в якому для кожної епохи відбувається обробка даних. Модель знаходиться в режимі навчання (`model.train()`), обчислюється втрата (помилка) через функцію `cross_entropy`, виконується зворотне поширення помилки та оновлення ваг моделі через оптимізатор. У процесі обчислюється точність кожному бітчі даних. Після закінчення кожної епохи виводяться метрики – втрата та точність.

Після завершення навчання модель зберігається у файл, щоб її можна використовувати для передбачень у майбутньому. Дана модель зберігається під назвою – `your_model_weights.pth`

Цей перший етап навчання включає в себе завантаження даних, налаштування моделі, а також безпосереднє навчання нейронної мережі з використанням заданих параметрів. Завдяки структурі цих файлів, вся системи має чітке розділення обов'язків: *train.py* відповідає за навчання, *YourModel.py* за архітектуру моделі, а *YourDataset.py* за обробку даних.

Після виконання першого етапу навчання нейронної мережі були отримані такі результати (рис.3.7):



```
(venv) PS E:\Diplom\ProjectFaceRecognition\src> python train.py
Epoch: 1/10, Loss: 8.894327163696289, Accuracy: 0.0
Epoch: 2/10, Loss: 8.618420600891113, Accuracy: 0.0
Epoch: 3/10, Loss: 8.60228443145752, Accuracy: 0.0002
Epoch: 4/10, Loss: 8.589906692504883, Accuracy: 0.0002
Epoch: 5/10, Loss: 8.545222282409668, Accuracy: 0.0002
Epoch: 6/10, Loss: 8.580212593078613, Accuracy: 0.0002
Epoch: 7/10, Loss: 8.554302215576172, Accuracy: 0.0002
Epoch: 8/10, Loss: 8.585613250732422, Accuracy: 0.0002
Epoch: 9/10, Loss: 8.562771797180176, Accuracy: 0.0002
Epoch: 10/10, Loss: 8.5570068359375, Accuracy: 0.0002
```

Рис.3.7. Результат першого навчання

З результатів можна побачити, що за всі 10 епох точність моделі майже не покращилася і залишилася на рівні 0.02%, а значення Loss (функція втрат, що показує точність моделі) зменшилося незначно з 8.89 до 8.55. Це вказує на кілька можливих проблем, які потребують уваги:

- 1) Занадто мала кількість даних: оскільки точність майже не змінюється, це може означати, що модель не отримує корисної інформації з

даних (мала кількість даних, невелика різноманітність осіб, їх положення на фотографії тощо);

2) Параметри моделі: модель може бути занадто простою для вирішення задачі класифікації облич, або ж параметри моделі (наприклад, кількість епох, швидкість навчання) не оптимальні для цієї задачі. Це також може бути показником того, що модель не здатна навчитися наданому набору даних через невідповідну архітектуру;

3) Можливість перенавчання або відсутність достатньої складності: навчання з фіксованими параметрами може призвести до відсутності поліпшення якості на наступних епохах, що вказує на модель не здатна адекватно узагальнювати дані.

На основі результатів першого навчання, можна зробити висновок, що модель вимагає подальшої оптимізації та доопрацювання. Це може включати зміну параметрів навчання, покращення архітектури моделі або більш ретельну перевірку якості та кількості вихідних даних.

Для того, щоб досягти кращих результатів після першого етапу, де точність залишилася на вкрай низькому рівні – прийнято рішення провести другий етап навчання.

На другому етапі планується внести певні зміни у параметри навчання та вдосконалити архітектуру моделі.

2. Другий етап навчання: цей етап є покращеною версією архітектури моделі та процесу тренування. Основні нововведення включають регуляризацию, нормалізацію пакетів, валідацію та моніторинг через TensorBoard:

- Файл YourModel.py: другий етап навчання. Оновлена структурна схема архітектури зображена на рис.3.8.

Основними відмінностями другого навчання є:

1) Batch Normalization: доданий шар нормалізації (BatchNorm2d) після кожної згорткової операції, що покращує збіжність мережі та стабільність градієнтів;

2) Dropout: шар Dropout додається після першого повнозв'язного шару, щоб запобігти перенавчанню. Під час навчання цей шар випадковим чином відключає (обнуляє) деякі нейрони, що допомагає моделі краще узагальнювати і не залежати від окремих ваг;

3) ReLU та BatchNorm: у кожному згортковому блоці спочатку застосовується ReLU, а потім BatchNorm, для стабілізації тренувального процесу.

Оновлена модель є складнішою і спрямована на поліпшення продуктивності за рахунок нормалізації та регуляризації. Вона включає в себе методи боротьби з перенавчанням.

YourModel.py, second training

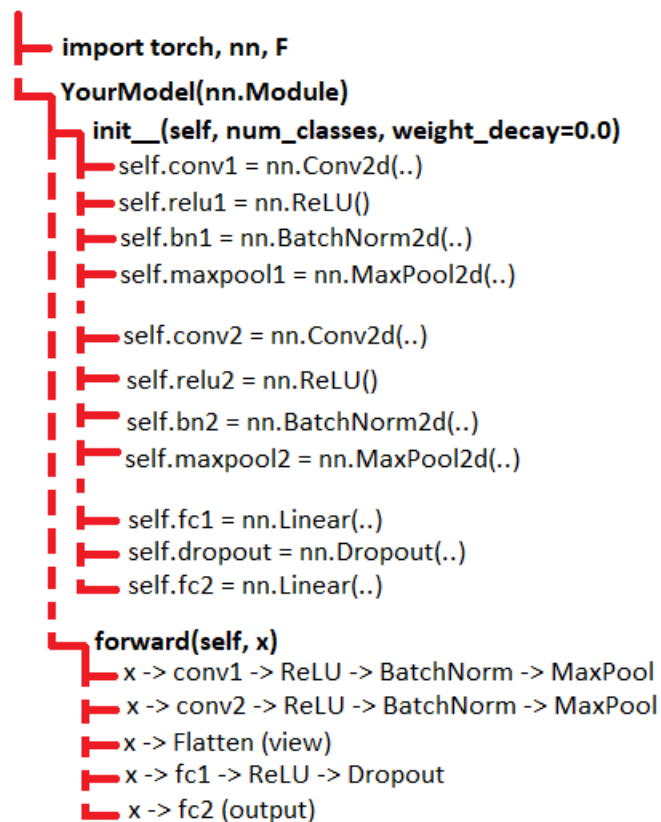


Рис.3.8. Структурна схема YourModel.py, second training

- Файл train.py: у другому навчанні, спочатку були внесені зміни для гіперпараметрів. У другому навчанні вони мають такі параметри: `batch_size = 64`; `learning_rate = 0.0001`; `epochs = 20`; `num_classes = 10001`;

`weight_decay = 0.001` (це новий параметр, який відповідає за регуляризацію вар). Структурна схема другого навчання зображена на рис.3.9

train.py, second training



Рис.3.9. Структурна схема train.py, second training

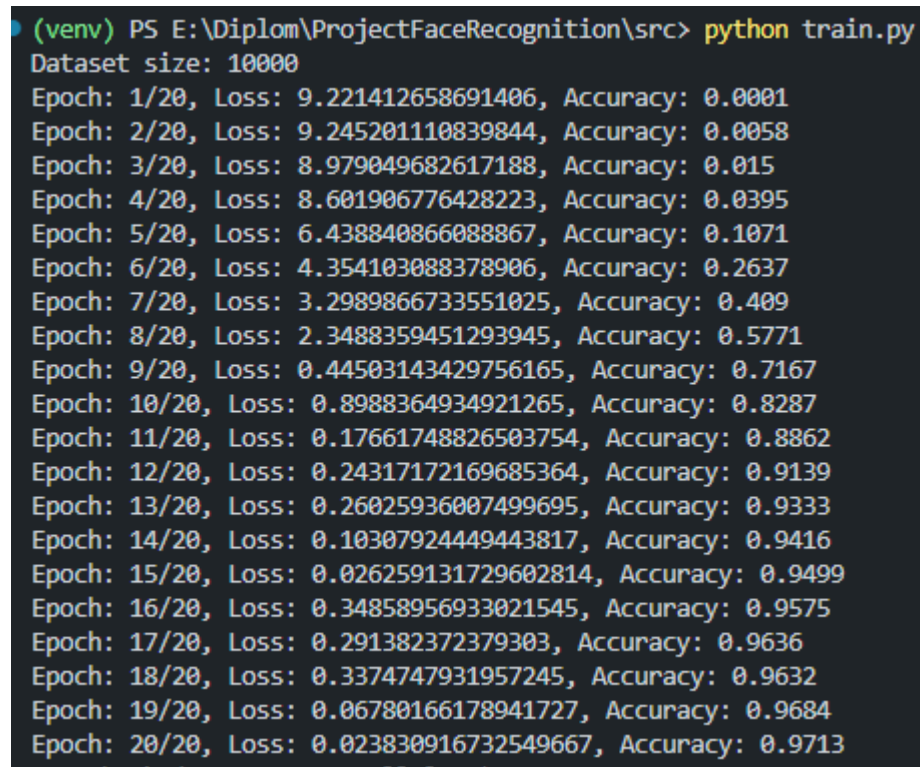
Основною задачею є проведення навчання нейронної мережі використовуючи нові методи та параметри. Цього разу додано навчальний і валідаційний набір даних, а також використовуються спеціальні перетворення для нормалізації зображень перед подачею їх до моделі.

Під час навчання модель тренується з використанням нових компонентів, таких як планувальник навчання (StepLR), який допомагає контролювати швидкість навчання на різних етапах. Для кожної епохи обчислюється втрата та точність, після чого модель перевіряється на валідаційних даних. В результаті використовується метод запису для логування метрик до TensorBoard.

Після завершення навчання ваги моделі зберігаються для подальшого використання.

Цей етап навчання покращує стійкість та точність моделі у задачах класифікації зображень.

Після виконання другого етапу навчання нейронної мережі були отримані результати, наведені на рис.3.10.



```
(venv) PS E:\Diplom\ProjectFaceRecognition\src> python train.py
Dataset size: 10000
Epoch: 1/20, Loss: 9.221412658691406, Accuracy: 0.0001
Epoch: 2/20, Loss: 9.245201110839844, Accuracy: 0.0058
Epoch: 3/20, Loss: 8.979049682617188, Accuracy: 0.015
Epoch: 4/20, Loss: 8.601906776428223, Accuracy: 0.0395
Epoch: 5/20, Loss: 6.438840866088867, Accuracy: 0.1071
Epoch: 6/20, Loss: 4.354103088378906, Accuracy: 0.2637
Epoch: 7/20, Loss: 3.2989866733551025, Accuracy: 0.409
Epoch: 8/20, Loss: 2.3488359451293945, Accuracy: 0.5771
Epoch: 9/20, Loss: 0.44503143429756165, Accuracy: 0.7167
Epoch: 10/20, Loss: 0.8988364934921265, Accuracy: 0.8287
Epoch: 11/20, Loss: 0.17661748826503754, Accuracy: 0.8862
Epoch: 12/20, Loss: 0.24317172169685364, Accuracy: 0.9139
Epoch: 13/20, Loss: 0.26025936007499695, Accuracy: 0.9333
Epoch: 14/20, Loss: 0.10307924449443817, Accuracy: 0.9416
Epoch: 15/20, Loss: 0.026259131729602814, Accuracy: 0.9499
Epoch: 16/20, Loss: 0.34858956933021545, Accuracy: 0.9575
Epoch: 17/20, Loss: 0.291382372379303, Accuracy: 0.9636
Epoch: 18/20, Loss: 0.3374747931957245, Accuracy: 0.9632
Epoch: 19/20, Loss: 0.06780166178941727, Accuracy: 0.9684
Epoch: 20/20, Loss: 0.023830916732549667, Accuracy: 0.9713
```

Рис.3.10. Результати другого навчання

З результатів другого навчання видно, що протягом 20 епох модель демонструє значне покращення точності. Початкова точність становила лише 0.0001%, але вже до кінця навчання досягла 97.13%. Це свідчить про те, що модель ефективно навчалась і змогла розпізнати особливості класів.

Значення Loss також зазнало істотних змін: воно зменшилось з 9.221 до 0.0238. Це вказує на те, що модель поступово покращує свої передбачення і зменшує помилки.

Порівнюючи з першим етапом, де точність майже не змінювалася, другий етап продемонстрував:

- 1) Покращення точності: за 20 епох точність зросла з 0.0001% до 97%, що є значним прогресом;
- 2) Зменшення втрат: значення Loss зменшилося з 9.221 до 0.0238, що вказує на ефективність навчання.

На основі цих результаті можна зробити висновок, що модель значно покращила свої характеристики порівняно з першим етапом. Це може свідчити про правильний вибір параметрів моделі, збільшення обсягу навчальних даних та вдосконалення архітектури.

На даному етапі, ця модель має хороший потенціал для успішного розпізнавання облич. Проте, незважаючи на успішні результати, існує ще декілька аспектів, які варто вдосконалити для досягнення ще вищих показників:

- 1) Доцільно провести більш тонке налаштування гіперпараметрів, таких як швидкість навчання, кількість епох та розмір батча, щоб знайти оптимальні значення для покращення якості навчання. Також можна переглянути архітектуру моделі – додавання нових шарів, зміна їх параметрів чи застосування інших функцій активації можуть допомогти моделі краще узагальнювати дані;
- 2) Важливим плюсом є перевірити якість навчальних даних, перед їх заміною, оскільки присутність шуму або неправильних міток може вплинути на результати.

Отже, третій етап стане ще однією можливістю вдосконалити модель, щоб спробувати підвищити її ефективність і забезпечити кращу здатність до навчання на нових даних.

3. Третій етап навчання: цей етап буде сфокусовано на більш тонке налаштування гіперпараметрів, оптимізацію та покращення архітектури, щоб спробувати досягти ще вищих результатів у класифікації облич:

- Файл `YourModel.py`: у третій версії моделі зроблено кілька змін у порівнянні з попередніми етапами, що спрямовані на покращення якості розпізнавання зображень. Структурна схема третьої архітектури зображена на рис.3.11.

У дану архітектуру додано третій згортковий шар, що збільшує здатність моделі розпізнавати складніші ознаки зображень. Перший згортковий шар обробляє вхідні зображення з трьома каналами (RGB) і використовує 32 фільтри. Другий шар збільшує кількість фільтрів до 64, а третій – до 128, що дозволяє моделі краще розпізнавати дрібні деталі зображення, такі як текстури або складні об'єкти, завдяки більшій глибині моделі.

Кожен згортковий шар супроводжується функцією активації ReLU, нормалізацією (BatchNorm2d) і операцією MaxPool для зменшення розмірності даних та обчислювальної складності. Це дозволяє не тільки зменшити розмір вхідних даних, але й покращує ефективність навчання, оскільки нормалізація допомагає уникнути проблеми з високими градієнтами.

У оновленій архітектурі повнозв'язний шар (fc1) має 512 нейронів, тоді як раніше їх було менше. Це дозволяє обробляти більше ознак після проходження через сверточні шари, що покращує здатність моделі до узагальнення та забезпечує кращу інтеграцію різноманітних ознак зображення для класифікації.

Важливою зміною є використання регуляризації через Dropout (0.5), який допомагає уникнути перенавчання, вимикаючи частину нейронів під час навчання. Це зменшує ймовірність того, що модель надмірно

підлаштовується під тренувальні дані, збільшуючи її стійкість до нових, невідомих даних і покращує її узагальнюючу здатність.

Завершується модель лінійним шаром (fc2), який відповідає за класифікацію на основі кількості класів. Вихід цього шару – це ймовірності, що зображення належить до того чи іншого класу, і таким чином, модель здатна робити точні прогнози для нових зображень, що не зустрічалися під час навчання.

YourModel.py, third training

```
import torch, nn, F

class YourModel(nn.Module):
    def __init__(self, num_classes, weight_decay=0.0):
        self.conv1 = nn.Conv2d(3, 32, kernel_size=5, stride=1, padding=2)
        self.relu1 = nn.ReLU()
        self.bn1 = nn.BatchNorm2d(32)
        self.maxpool1 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2)
        self.relu2 = nn.ReLU()
        self.bn2 = nn.BatchNorm2d(64)
        self.maxpool2 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.conv3 = nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=2)
        self.relu3 = nn.ReLU()
        self.bn3 = nn.BatchNorm2d(128)
        self.maxpool3 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.fc1 = nn.Linear(128 * 28 * 28, 512)
        self.dropout = nn.Dropout(0.5)
        self.fc2 = nn.Linear(512, num_classes)

    def forward(self, x):
        x -> conv1 -> ReLU -> BatchNorm -> MaxPool
        x -> conv2 -> ReLU -> BatchNorm -> MaxPool
        x -> conv3 -> ReLU -> BatchNorm -> MaxPool
        x -> Flatten (view)
        x -> fc1 -> ReLU -> Dropout
        x -> fc2 (output)
```

Рис.3.11. Структурна схема YourModel.py, third training

Отже, збільшення кількості фільтрів у сверточних шарах та додавання третього шару дозволяє моделі витягати більше ознак із зображень, що сприятиме кращому розпізнаванню навіть у складних умовах. Використання Dropout допоможе уникнути перенавчання, роблячи модель більш універсальною та здатною працювати з новими даними.

- Файл `YourDataset.py`: у даній версії внесено кілька оновлень, які покращують роботу з даними і роблять модель стійкішою до проблем із завантаженням зображень. Оновлена структурна схема зображена на рис.3.12.

YourDataset.py, third training

```
import os, pandas, Dataset, transforms, Image
YourDataset(Dataset)
    init__(self, csv_file, root_dir, transform=None)
        self.annotations = pd.read_csv(csv_file)
        self.root_dir = root_dir
        self.transform = transform
    len__(self)
        return len(self.annotations)
    getitem__(self, idx)
        img_path = os.path.join(..)
        try: image = Image.open(..).convert("RGB")
        except Exception: image = Image.new("RGB", (224, 224), (225, 225, 225))
        label = int(self.annotations.iloc[idx, 1])

        if self.transform:
            image = self.transform(image)
        return image, label
```

Рис.3.12. Структурна схема файлу `YourDataset, third training`

Були внесені такі оновлення в метод `getitem`:

- 1) обробка винятків при завантаженні зображень: додано механізм, який обробляє помилки при відкритті зображень. Якщо файл пошкоджений або його немає, замість зупинки процесу створюється пусте зображення розміром 224x224 з сірим фоном. Це підвищує стійкість системи до проблем із файлами;

2) Повернення даних: незмінно, метод повертає зображення та його мітку, але тепер враховується можливість повернення “пустого” зображення у випадку помилки.

Отже, додавання обробки винятків дозволяє уникнути проблем із якістю даних – процес навчання не перерветься через пошкоджене або відсутнє зображення. Це також позитивно вплине на стабільність моделі під час навчання, оскільки відсутність збоїв у датасеті дозволить моделі краще навчатися на наявних даних.

Крім того, застосування трансформації робить модель більш гнучкою до варіацій у зображеннях (наприклад, різні розміри або формати), що підвищує її здатність узагальнювати знання і працювати на різноманітних даних.

- Файл `train.py`: у цьому етапі додано кілька оновлень, що покращують процес навчання і дозволяють ефективніше відстежувати прогрес моделі. Оновлена структурна схема зображена на рис.3.13.

В третьому етапі навчання були знову змінені гіперпараметри. У третьому навчанні вони мають такі параметри: `batch_size = 64`; `learning_rate = 0.00001`; `epochs = 25`; `num_classes = 15001`; `weight_decay = 0.001`. Параметр `batch_size` не змінюється, оскільки його зменшення не мало б впливу на якість навчання моделі, враховуючи вже досягнуту стабільність. Водночас збільшення `batch_size` не можливо через апаратні обмеження – пристрій не зміг би обробляти великі обсяги даних на кожній ітерації без значного впливу на продуктивність.

Що стосується `learning_rate`, вирішено знизити його до `0.00001` для того, щоб модель могла більш точно коригувати ваги на пізніх етапах навчання. Висока швидкість навчання на цьому етапі могла б призвести до нестабільних результатів, тому поступове зменшення кроку дозволяє уникнути різних змін у процесі оптимізації, що покращує точність моделі.

Далі, розширено роботу з даними завдяки аугментації, яка підвищує здатність моделі до узагальнення, додаючи варіативність до існуючих даних

(випадкові обертання, зміни кольору тощо). Крім цього, введено планувальник StepLR для зменшення швидкості навчання на пізніх етапах, що дозволяє більш плавно коригувати ваги моделі для підвищення точності.

Ще одним покращенням стало введення валідації після кожної епохи, що дозволяє відстежувати перенавчання та забезпечує більш точну оцінку узагальнення моделі на нових даних. Для моніторингу процесу навчання також додано підтримку TensorBoard, що дає змогу візуалізувати динаміку втрат і точності.

Завершується процес навчання збереженням ваг моделі, що забезпечує її подальше використання або відновлення для майбутніх експериментів.



Рис.3.13. Структурна схема train.py, third training

Ці нововведення покращують контроль над процесом навчання та забезпечують кращу ефективність і точність передбачень.

Після виконання третього етапу навчання нейронної мережі були отримані такі результати (рис.3.14):

```
(venv) PS E:\Diplom\ProjectFaceRecognition\src> python train.py
Dataset size: 15000
Epoch: 1/25, Loss: 9.627565383911133, Accuracy: 0.0
Epoch: 2/25, Loss: 9.641749382019043, Accuracy: 0.0018666666666666666
Epoch: 3/25, Loss: 9.180657386779785, Accuracy: 0.0062
Epoch: 4/25, Loss: 8.29540729522705, Accuracy: 0.0224
Epoch: 5/25, Loss: 6.6487135887146, Accuracy: 0.083
Epoch: 6/25, Loss: 4.8320817947387695, Accuracy: 0.2789333333333333
Epoch: 7/25, Loss: 3.139331579208374, Accuracy: 0.46986666666666665
Epoch: 8/25, Loss: 1.8535927534103394, Accuracy: 0.6526666666666666
Epoch: 9/25, Loss: 1.0319987535476685, Accuracy: 0.7785333333333333
Epoch: 10/25, Loss: 1.040758490562439, Accuracy: 0.8596
Epoch: 11/25, Loss: 0.8387059569358826, Accuracy: 0.9083333333333333
Epoch: 12/25, Loss: 0.6721355319023132, Accuracy: 0.9225333333333333
Epoch: 13/25, Loss: 0.5564002394676208, Accuracy: 0.9384666666666667
Epoch: 14/25, Loss: 0.14728090167045593, Accuracy: 0.9392
Epoch: 15/25, Loss: 0.5681795477867126, Accuracy: 0.9497333333333333
Epoch: 16/25, Loss: 0.20465123653411865, Accuracy: 0.9545333333333333
Epoch: 17/25, Loss: 0.8572718501091003, Accuracy: 0.9582666666666667
Epoch: 18/25, Loss: 0.05398137494921684, Accuracy: 0.9636666666666667
Epoch: 19/25, Loss: 0.17441676557064056, Accuracy: 0.9640666666666666
Epoch: 20/25, Loss: 0.24647696316242218, Accuracy: 0.9660666666666666
Epoch: 21/25, Loss: 0.10150567442178726, Accuracy: 0.9686666666666667
Epoch: 22/25, Loss: 0.32448330521583557, Accuracy: 0.9700666666666666
Epoch: 23/25, Loss: 0.07491333782672882, Accuracy: 0.9692666666666667
Epoch: 24/25, Loss: 0.16059735417366028, Accuracy: 0.9703333333333334
Epoch: 25/25, Loss: 0.07864189147949219, Accuracy: 0.972
Validation Accuracy: 1.000
```

Рис.3.14. Результат третього навчання

З результатів третього навчання видно, що модель продемонструвала покращення точності. Основні показники:

- точність: модель досягла максимальної (як для останнього навчання) точності 97.2% на завершальному етапі навчання. Це свідчить про те, що модель здатна правильно класифікувати 97.2% зразків, що є покращенням порівняно з другим етапом, де точність була трохи нижчою — на 0.072% менше. Цей прогрес підтверджує, що нові методи аугментації даних та оптимізації навчання позитивно вплинули на загальну продуктивність моделі;
- значення Loss: величина Loss знизилась до 0.0786, що свідчить про зменшення кількості помилок, які модель робила під час навчання, та про її здатність точніше прогнозувати результати для заданих класів.

У порівнянні з другим етапом, де максимальна точність становила 97.13%, третій етап продемонстрував:

- 1) покращення точності: точність зросла до 97.2%, що свідчить про ефективнішу генералізацію моделі;
- 2) покращення втрат: значення Loss знизилося до 0.0786, що вказує на вищу якість передбачень у порівнянні з попереднім етапом.

На основі цих результатів можна стверджувати, що модель покращила свої характеристики порівняно з другим етапом. Запроваджені зміни в архітектури, такі як аугментація даних та нові оптимізаційні моделі, сприяли досягненню вищих показників точності. Ці результати підкреслюють правильність вибору параметрів моделі та вдалий підхід до навчання, що забезпечує кращу якість передбачень.

Після виконання третього етапу навчання моделі, прийнято рішення зупинити навчання. Це обумовлено кількома важливими факторами:

- 1) уникнення перенавчання: зі збільшенням кількості епох існує ризик перенавчання моделі, коли вона починає вивчати не тільки корисні патерни, а й шум у даних. Це призводить до погіршення загальної продуктивності на нових, невідомих даних. Тому, з метою збереження якості моделі, прийнято рішення завершити навчання на третьому етапі;
- 2) обмеження апаратних ресурсів: під час навчання з великою кількістю даних, пристрій, на якому здійснюється виконання проекту, не справлявся з навантаженням, що викликало значні проблеми з продуктивністю. Модель починала демонструвати затримки в обробці, а в окремих випадках пристрій повністю зависав та міг вимкнутися;
- 3) час навчання: кожен етап навчання вимагав значну затрату часу. Наприклад, перше навчання тривало лише, приблизно 1.5 години, друге – понад 9-10 годин, а третє – близько 14-15 годин. Цей фактор також вплинув на рішення обмежити кількість навчань;

4) неможливість розподілу навчання: відсутня можливість використовувати кілька графічних процесорів для розподілу навантаження. Ноутбук починав сильно перегріватися, що призводило до критичних ситуацій, коли він просто вимикався через перегрів.

Таким чином, обмеження до трьох циклів навчань обумовлено не лише апаратними обмеженнями, а й міркуваннями для збереження якості моделі. Зокрема, прагнення уникнути перенавчання та забезпечити стабільну роботу пристрою стала основними причинами для прийняття цього рішення. Цей підхід дозволив досягти оптимальних результатів за обмеженого часу та ресурсів, що є важливим аспектом в умовах обмеженого апаратного забезпечення.

3.2. Реалізація модуля розпізнавання осіб.

У попередньому параграфі розглянуто процес навчання моделі для розпізнавання осіб, що дало змогу досягти значних успіхів у її точності та ефективності. Тепер переходимо до наступного етапу – реалізації модуля розпізнавання осіб, в рамках якого будемо використовувати навчену модель “your_model_weights.pth” в таких створених файлах: YourDatabase.py та combine.py.

Для ефективної роботи модуля розпізнавання осіб важливо створити інтуїтивно зрозумілий і зручний інтерфейс. Візуальна частина програми забезпечить користувачам легкий доступ до всіх функцій і налаштувань, а також надасть можливість швидко отримувати результат в реальному часі.

Почнемо зі створення структури програми (рис.3.15). Як вже сказано раніше, інтерфейс програми розроблено простим і інтуїтивно зрозумілим для користувача. Він складається з наступних елементів:

1. Головне вікно “Face Recognition App”:
 - Кнопка “Start Recognition” – запускає процес розпізнавання облич у режимі реального часу. При натисканні кнопки програма починає зчитувати

дані з камери і аналізувати кадри пошуку облич. Після успішного виявлення обличчя програма намагається ідентифікувати його та відобразити відповідну інформацію у вікні “Employee Inform”;

- Кнопка “Save Photo” – ця функція використовується для збереження фотографії розпізнавання обличчя за певним шляхом.

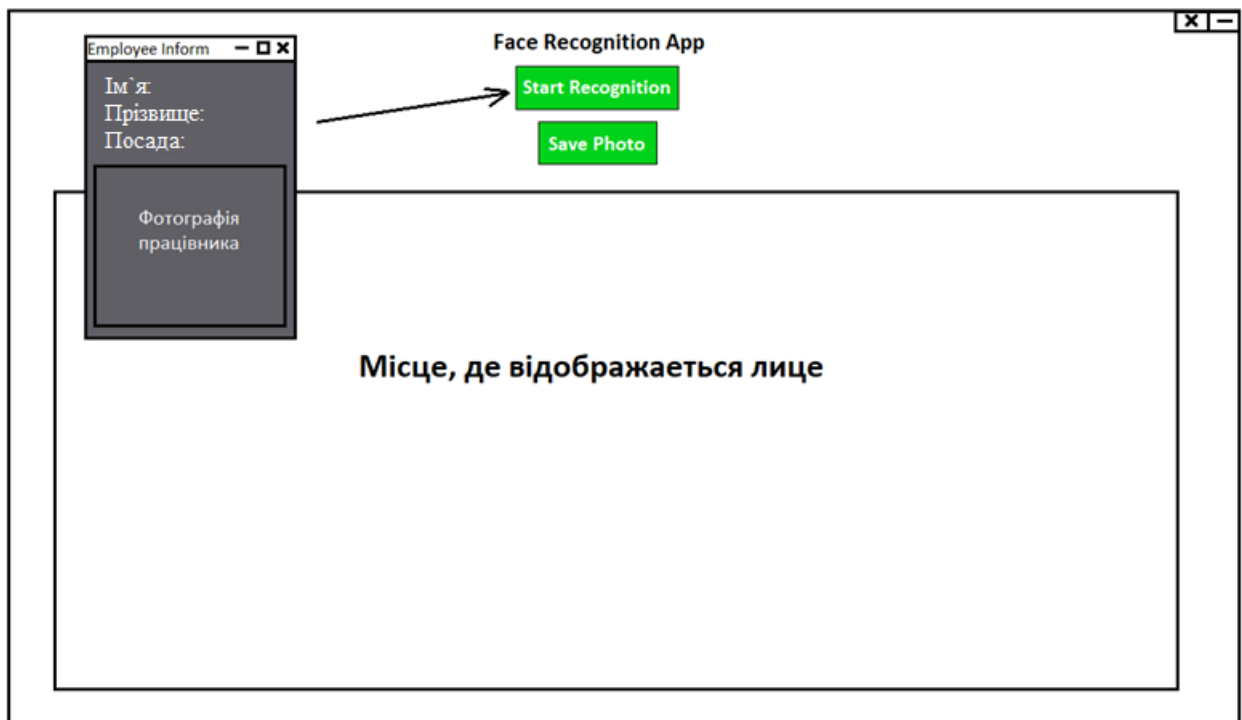


Рис.3.15. Інтерфейс програми

2. Вікно “Employee Inform” (з’являється при натисканні кнопки “Start Recognition”):

- Ім’я, Прізвище, Посада – у цих полях буде відображатися інформація про співробітника, чиє обличчя розпізнано;
- Фотографія працівника – тут буде показане фото працівника, відповідне до розпізнаного обличчя. Це необхідно для візуального підтвердження результатів розпізнавання.

3. Місце, де відображається обличчя – основна область, де відображатиметься зображення з камери та результат розпізнавання обличчя

в режимі реального часу. Ця область потрібна для візуального контролю та підтвердження, що процес розпізнавання працює належним чином.

Майбутня програма повинна працювати таким чином:

1. Запуск розпізнавання: користувач натискає кнопку “Start Recognition”, після чого програма починає аналізувати відеопотік з камери. Як тільки обличчя буде виявлено, програма спробує ідентифікувати його за допомогою завантажених моделей та даних. Якщо розпізнавання буде успішним, інформація про співробітника з’явиться у вікні “Employee Inform”, у іншому випадку – відобразиться попередження.

2. Збереження фотографії: користувач може натиснути кнопку “Save Photo”, якщо потрібно зберегти зображення розпізнаного обличчя. Ця функція може використовуватися для забезпечення безпеки – наприклад, для ведення бази зображень облич, що з’являються в кадрі, або для подальшого аналізу. Усі збереження фотографії зберігатимуться за вказаним шляхом для подальшого використання.

3. Відображення обличчя: в основній області програми відображатиметься зображення з камери, що дозволить користувачеві бачити все, що відбувається в режимі реального часу. Якщо обличчя розпізнано, воно буде виділене або позначене в цій області.

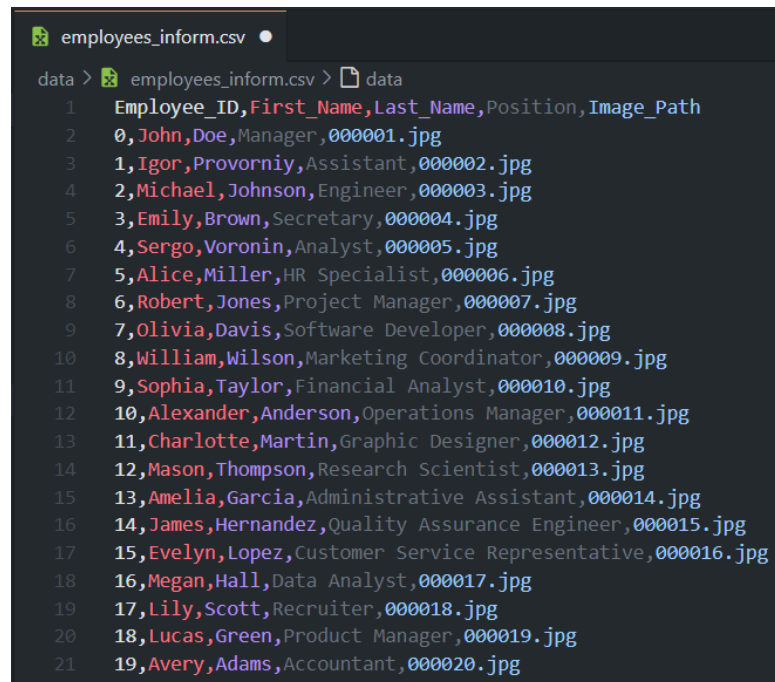
Таким чином, інтерфейс програми є простим і зрозумілим, надаючи всі необхідні функції для розпізнавання облич і збереження результатів.

Перед початком розробки програми, до проекту додано файл: `employees_inform.csv` (рис.3.16).

Створений файл слугує базою даних для зберігання інформації про співробітників організації. Кожен рядок у файлі містить наступні поля:

- `Employee_ID`: унікальний ідентифікатор співробітника;
- `First_Name`: ім’я співробітника;
- `Last_Name`: прізвище співробітника;
- `Position`: посада співробітника в організації;

- `Image_Path`: шлях до зображення обличчя співробітника, що використовується для розпізнавання.



```

employees_inform.csv
data > employees_inform.csv > data
1 Employee_ID,First_Name,Last_Name,Position,Image_Path
2 0,John,Doe,Manager,000001.jpg
3 1,Igor,Provorniy,Assistant,000002.jpg
4 2,Michael,Johnson,Engineer,000003.jpg
5 3,Emily,Brown,Secretary,000004.jpg
6 4,Sergo,Voronin,Analyst,000005.jpg
7 5,Alice,Miller,HR Specialist,000006.jpg
8 6,Robert,Jones,Project Manager,000007.jpg
9 7,Olivia,Davis,Software Developer,000008.jpg
10 8,William,Wilson,Marketing Coordinator,000009.jpg
11 9,Sophia,Taylor,Financial Analyst,000010.jpg
12 10,Alexander,Anderson,Operations Manager,000011.jpg
13 11,Charlotte,Martin,Graphic Designer,000012.jpg
14 12,Mason,Thompson,Research Scientist,000013.jpg
15 13,Amelia,Garcia,Administrative Assistant,000014.jpg
16 14,James,Hernandez,Quality Assurance Engineer,000015.jpg
17 15,Evelyn,Lopez,Customer Service Representative,000016.jpg
18 16,Megan,Hall,Data Analyst,000017.jpg
19 17,Lily,Scott,Recruiter,000018.jpg
20 18,Lucas,Green,Product Manager,000019.jpg
21 19,Avery,Adams,Accountant,000020.jpg

```

Рис.3.16. Файл бази даних співробітників

Таким чином, файл містить важливу інформацію для ідентифікації співробітників у процесі розпізнавання облич та може бути використаний для подальшої обробки даних в програмі.

Після формування структури програми та розгляду функціоналу проекту можемо переходити до розробки та опису програмного коду, реалізованого у файлах `YourDatabase.py` та `combine.py`, де буде показано, як працюють модулі для зберігання даних і обробки результатів розпізнавання:

- Файл `YourDatabase.py`: код, у даному файлі, відповідає за роботу з базою даних облич, обробку зображень, а також за завантаженням та використання попередньо натренованої моделі для порівняння облич. Структурна схема даного файлу зображена на рис.3.17.

Він складається з двох основних класів `EmployeeInfo` та `FaceDatabase`:

- 1) Клас `EmployeeInfo`: цей клас зберігає інформацію про співробітника, зокрема його `employees_id`, `first_name`, `last_name` та `position`. Він

допомагає структурувати дані про співробітників для подальшого використання в системі розпізнавання облич;

2) Клас FaceDatabase: основний клас, який відповідає за роботу з базою даних облич та натренованою моделлю. Він містить кілька ключових методів:

- Ініціалізація класу: під час ініціалізації класу відбувається перевірка, чи існує шлях до натренованої моделі (`model_weight_path`). Якщо цей шлях існує, модель `YourModel` завантажується з файлу `your_model_weights.pth` і переводиться у режим оцінювання.

Також завантажується CSV з інформацією про співробітників та зображення облич із заданої директорії, а для розпізнавання облич завантажується класифікатор каскадів Хаара (`CascadeClassifier`) для визначення облич на зображеннях;

- `preprocess_image(self, image)`: метод для попередньої обробки зображення. Він застосовує трансформації до зображення для підготовки його до подачі в модель;

- `get_embeddings(self, employee_id)`: за допомогою цього методу генерується ембедінг (векторне представлення) обличчя співробітника. Зображення завантажується з вказаного шляху, обробляється, після чого модель генерує ембедінги;

- `get_most_similar_employee(self, query_embeddings)`: цей метод порівнює отримані ембедінги із базою ембедінгів усіх співробітників. Використовується функція обчислення косинусної відстані для знаходження найбільш схожого обличчя. Косинусна відстань між двома векторами A і B обчислюється за формулою:

$$\text{cosin_distance}(A, B) = 1 - \frac{A * B}{\|A\| \|B\|}$$

де:

$A * B$ – скалярне добутку векторів A і B ;

$\|A\|$ та $\|B\|$ – норми (довжини) векторів A і B відповідно.

Метод мінімізує косинусне відстань між ембедінгами запиту (query_embeddings) та ембедінгами у базі даних для пошуку найбільш схожого обличчя, відповідно визначаючи потрібного співробітника;

- `check_model(self)`: метод для перевірки роботи моделі на ембедінгах всіх співробітників. Він обчислює ембедінги для кожного обличчя та виводить їх на екран;

- `check_face_detection(self)`: метод для перевірки детекції облич на зображеннях співробітників із бази. Якщо обличчя не знаходиться, виводиться повідомлення про це;

- `check_embeddings_distance(self)`: метод для перевірки дистанції між ембедінгами облич усіх співробітників;

- `check_most_similar_employee(self)`: використовує попередні методи для обчислення схожості між ембедінгами та пошуку найбільш схожого співробітника.

Файл `YourDatabase.py` буде підключатися до файлу `combine.py`, щоб забезпечити доступ до натренованої моделі та бази ембедінгів. У `combine.py` через клас `FaceDatabase` завантажується модель, а потім обчислюються ембедінги облич, які порівнюються з наявними даними про співробітників. Це дозволяє у реальному часі розпізнавати обличчя за допомогою камери та відображати результати на інтерфейсі програми.

Таким чином, код у файлі `YourDatabase` є ключовим компонентом для обробки облич та роботи з моделлю, тоді як `combine` буде відповідати за інтеграцію цієї логіки в інтерфейс програми та буде забезпечувати взаємодію користувача з системою.

- Файл `combine.py`: є центральним елементом програми, що відповідає за інтеграцію всіх модулів, пов'язаних із розпізнаванням облич, та реалізацію основної структури програми. Він забезпечує обробку даних, отриманих з модуля `YourDatabase.py`, та реалізує логіку для виконання розпізнавання облич за допомогою навченої моделі. Структурна схема файлу зображена на рис.3.18

Основними компонентами цього файлу є:

1) FaceRecognitionApp: основний клас програми, що відповідає за ініціалізацію графічного інтерфейсу, управління відеопотоком та виконання розпізнавання облич;

○ $\text{init}(\text{self}, \text{master}, \text{face_database})$: конструктор класу, який ініціалізує основні параметри: вікно GUI, базу даних облич і елементи інтерфейсу, такі як кнопки та мітки. Також запускає метод $\text{update}()$ для регулярного оновлення відеопотоку;

2) $\text{start_recognition}(\text{self})$: цей метод включає в себе конвертацію зображення, виявлення обличчя, отримання ембедінгів та їх порівняння з базою даних. Основними етапами є:

1. Виявлення обличчя. Метод спочатку виявляє обличчя на кадрі за допомогою детектора облич. Для кожного кадру I детектор повертає область, де знаходиться обличчя:

$$\text{face_region} = f_{\text{detector}}(I);$$

2. Перетворення в відтінки сірого. Зображення перетворюється у відтінки сірого за допомогою формули:

$$I_{\text{gray}} = 0.299 * I_R + 0.587 * I_G + 0.114 * I_B$$

де I_R, I_G, I_B – значення каналів RGB;

3. Отримання ембедінгів. Ембедінг для області обличчя отримується за допомогою функції:

$$E = f_{\text{embedding}}(\text{face_region})$$

4. Порівняння ембедінгів. Отриманий ембедінг порівнюється з ембедінгами з база даних за допомогою косинусної відстані:

$$\text{cosine_distance}(E, E_i) = 1 - \frac{E * E_i}{\|E\| \|E_i\|}.$$

де E_i – ембедінг кожного співробітника з бази даних. Співробітник з мінімальним значенням косинусної відстані вважається найбільш схожим.

Метод проходить всі ці етапи для визначення найбільш схожого обличчя з бази даних;

3) `show_frame(self, frame, faces)`: метод, що відображає оброблений кадр з відеопотоку. Він конвертує кадр в формат RGB, малює прямокутники навколо виявлених облич і оновлює canvas, щоб показати новий кадр;

4) `update(self)`: метод, який відповідає за регулярне читання кадрів з камери, виявлення облич і показ оброблених кадрів. Викликається безперервно, поки програма працює;

5) `show_message(self, message, color)`: метод, що відповідає за оновлення тексту на мітці повідомлень у графічному інтерфейсі. Дозволяє відображати статус процесу розпізнавання або інші повідомлення користувачу;

6) `show_employee_info(self, employee_info)`: метод, що створює нове вікно для відображення детальної інформації про співробітника, який був розпізнаний. Інформація включає ім'я, прізвище та посаду;

7) `save_photo(self)`: метод, що дозволяє зберегти поточний кадр з відеопотоку (може бути корисно для збереження зображення розпізнаного обличчя);

8) `close_app(self)`: метод, що відповідає за закриття програми. Він звільняє ресурси відео-захоплення та закриває вікно програми;

9) `__name__ == "__main__"`: ініціалізує шляхи до необхідних файлів та створює об'єкт класу `FaceDatabase`, потім запускає графічний інтерфейс програми. Цей блок відповідає за запуск усієї програми.

У проекті використовується навчання з учителем на основі функції втрат крос-ентропії для багато-класової класифікації осіб. Основні етапи включають:

1. Ініціалізація мережі та завантаження даних:

- Дані зображень завантажуються та проходять попередню обробку включаючи зміну розміру, випадкове обертання, горизонтальне відображення та нормалізацію;

- Для кожного зображення призначається мітка, що відповідає ID співробітника.

2. Процес навчання:

- кожне зображення проходить через мережу. Спочатку згорткові шари виділяють особливості обличчя, а потім повнозв'язні шари перетворюють їх у ембедінги;

- на етапі прямого поширення (forward pass) модель генерує вихідний вектор для кожного зображення;

- помилка вихідного значення обчислюється за допомогою функції втрат крос-ентропії, що вимірює різницю між передбаченими та справжніми мітками.

3. Зворотне поширення помилки та оновлення ваг:

- використовується метод градієнтного спуску та оптимізатор Adam для корекції ваг мережі. Процес повторюється на кожній епосі навчання з метою мінімізації помилки класифікації.

4. Відстеження та налаштування гіперпараметрів:

- регуляризація та зниження швидкості навчання налаштовуються на основі даних про помилку та точність, які фіксуються через TensorBoard.

Після навчання модель перетворює кожне зображення обличчя в ембедінг (вектор фіксованої довжини, що представляє унікальні риси обличчя). Для порівняння ембедінгів використовується евклідова відстань (L2), що обчислюється за допомогою функції `pairwise_distance`. Ембедінг характеризується наступними параметрами:

1. Формула евклідової відстані:

- обчислюється як

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2},$$

де x і y – два ембедінг-вектори обличчя, а i – координати векторів.

- У коді, файлу `YourDatabase.py`, це реалізовано через:

```
# Обчислюємо евклідову відстань між вкладеннями
distance = torch.nn.functional.pairwise_distance(query_embeddings, target_embeddings).item()
```

YourDatabase.py

```

import torch, YourModel, transforms, os, pd, Image, cv2, F

EmployeeInfo class
    init__(self, employee_id, first_name, last_name, position)
        self.employee_id = employee_id
        self.first_name = first_name
        self.last_name = last_name
        self.position = position

FaceDatabase class
    init__(self, model_weight_path, csv_file_path, images_folder_path, num_classes)
        if model_weight_path exists
            self.model = YourModel(..)
            self.model.load_state_dict(torch.load(..))
            self.model.eval()
        self.employee_info = pd.read_csv(csv_file_path)
        self.images_folder_path = images_folder_path
        self.face_cascade = cv2.CascadeClassifier(..)
        self.embeddings_dict = {employee_id: self.get_embeddings(..)} for all employees

    preprocess_image(self, image)
        transform = transforms.Compose([..])
        return transform(image)

    get_embeddings(self, employee_id)
        image_path = os.path.join(..)
        image = Image.open(..).convert("RGB")
        image_tensor = self.preprocess_image(..)
        embeddings = self.model(image_tensor)
        embeddings = F.normalize(embeddings, p=2, dim=1)
        return embeddings

    get_employee_info(self, employee_id)
        return EmployeeInfo(..)

    get_most_similar_employee(self, query_embeddings)
        for each employee in self.embeddings_dict
            distance = torch.nn.functional.pairwise_distance(..)
            if distance < min_distance
                most_similar_employee_id = employee_id
        return most_similar_employee_id

    check_model(self)
        for each employee in employee_info
            self.get_embeddings(..)
            print embeddings

    check_face_detection(self)
        for each employee in employee_info
            image_path = os.path.join(..)
            image = cv2.imread(image_path)
            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            faces = self.face_cascade.detectMultiScale(..)
            if no faces
                print No face detected
            else
                print Face detected

    check_embeddings_distance(self)
        for each employee pair in employee_info
            distance = torch.nn.functional.pairwise_distance(..)
            print distance

    check_most_similar_employee(self)
        for each employee in employee_info
            embeddings = self.get_embeddings(..)
            most_similar_employee_id = self.get_most_similar_employee(..)
            print most similar employee

Create instance of FaceDatabase
    face_database = FaceDatabase(..)
    face_database.check_model()
    face_database.check_face_detection()
    face_database.check_embeddings_distance()
    face_database.check_most_similar_employee()

```

Рис.3.17. Структурна схема файлу YourDatabase

2. Порогове значення:
 - отримана відстань порівнюється з порогом. Якщо відстань менша за порогове значення, обличчя вважається однаковими; якщо більша – різними.
3. Нормалізація ембедінгів:
 - нормалізація (через `F.normalize`) допомагає зробити вектори одиничними, що полегшує порівняння та робить відстань більш стійкою до змін масштабу.

Ці методи у комбінації забезпечують точне та стабільне розпізнавання облич у системі.

combine.py

```

import tkinter, Image, ImageTk, ImageDraw, FaceDatabase, cv2, torch, os, time, numpy

class FaceRecognitionApp:
    def __init__(self, master, face_database):
        self.master, self.face_database, self.video_capture = master, face_database, cv2.VideoCapture(0)
        # GUI elements: label, start_button, save_button, message_label, canvas
        self.update()

    def start_recognition(self):
        ret, frame = self.video_capture.read()
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = self.face_database.face_cascade.detectMultiScale(...)
        for each face in faces:
            # process face (ROI)
            # get embeddings
            employee_id = self.face_database.get_most_similar_employee(embeddings)
            # show message based on employee_id
            # draw rectangle around face

    def show_frame(self, frame, faces):
        # convert frame to RGB
        # draw rectangles around faces
        # update canvas

    def update(self):
        # read frame from video_capture
        # detect faces
        # show frame with detected faces

    def show_message(self, message, color):
        # update message_label text

    def show_employee_info(self, employee_info):
        # create new window for employee details

    def save_photo(self):
        # save the current photo frame

    def close_app(self):
        # release video_capture resources and close app

if __name__ == "__main__":
    # Initialize: csv_file_path, images_folder_path, model_weight_path, your_num_classes
    face_database = FaceDatabase(csv_file_path, images_folder_path, model_weight_path, your_num_classes)
    # Define sharpening_kernel for image preprocessing
    root = tk.Tk(), app = FaceRecognitionApp(root, face_database)
    root.protocol("WM_DELETE_WINDOW", app.close_app), root.mainloop()
  
```

Рис.3.18. Структурна схема combine.py

Загалом, у `combine.py` фактично будується структура всієї програми, що була описана на схемі (рис.3.14):

1. Ініціалізуємо та завантажуюмо модель з базою ембедінгів у `YourDatabase.py`;
2. Кожен кадр з камери обробляється та використовується для виявлення облич;
3. Виявлені обличчя конвертуються в ембедінги за допомогою моделі, що була збережена;
4. Ці ембедінги порівнюються з базою даних, щоб знайти найбільш схожого співробітника;
5. Результат (ім'я, прізвище, позиція) відображаються в режимі реального часу.

Таким чином, `combine.py` є головним файлом для реального розпізнавання облич, що об'єднує всі раніше побудовані модулі (особливо `YourDatabase.py`) і дозволяє виконувати ідентифікацію співробітників на основі відеопотоку з камери.

Описані функції і структура цих файлів демонструють, як відбувається взаємодія між модулями, обробка відеопотоку з камери та ідентифікація співробітників. Залишається лише реалізувати та налаштувати систему, щоб підготувати її до подальшого тестування, яке буде проведено у наступному розділі.

3.3. Налаштування та тестування запропонованої системи.

Після формування схеми програми, реалізації структурних схем головних файлів та подальшого написання коду можемо приступити до тестування. Весь процес розробки проводився у віртуальному середовищі (`venv`), що дозволило ізолювати залежності проекту та забезпечити стабільність роботи програми.

Тестування стане важливим етапом, оскільки буде перевірена функціональність системи на практиці. У цьому розділі проведемо тести на розпізнання осіб за зображеннями, щоб переконатися, що програма здатна правильно ідентифікувати співробітників на основі зображень із бази даних.

Перевіримо систему на трьох етапах: запуск програми без активного розпізнавання облич; реакцію програми у випадку, коли обличчя не знайдено; перевірку ідентифікації особи із бази даних. Тепер можемо переходити до тестування:

1. Перший етап: запусимо програму, щоб оцінити інтерфейс та його функціональні елементи без активного розпізнавання облич. Ознайомитися з результатом можна на рис.3.19.

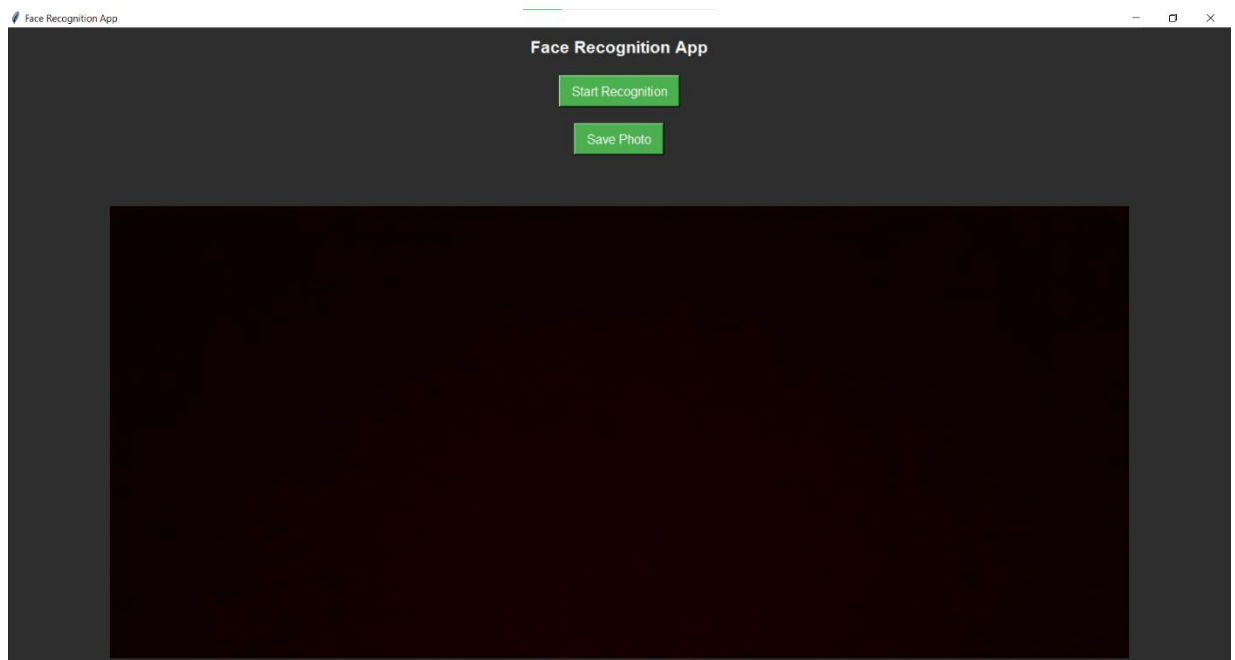


Рис.3.19. Інтерфейс програми Face Recognition App

На даній фотографії представлений готовий інтерфейс програми, який зроблено по структурній схемі програми. Зараз він знаходиться у стані очікування, коли система ще не почала обробляти відеопотік. Камера закрита, тому жодне зображення або обличчя не захоплені:

- 1) інтерфейс програми містить дві основні кнопки:
 - Start Recognition – для запуску процесу розпізнавання облич із відеопотоку;

- Save Photo – для збереження поточного кадру з камери, якщо це потрібно;

2) область для відеопотоку: відображається у вигляді порожнього вікна, де зазвичай показується кадр з камери;

3) відсутність віна Employees Inform вказує на те, що жоден співробітник ще не був розпізнаний, і система знаходиться в режимі очікування для початку процесу.

Цей рисунок демонструє стандартний стан програми, коли вона готова до обробки, але ще не отримала жодних даних для аналізу. Програма чекає на дію користувача, щоб розпочати розпізнавання облич за допомогою натискання кнопки “Start Recognition”.

2. Другий етап: перевіряємо реакцію програми на відсутність облич у кадрі. Результат даного етапу зображено на рис.3.20.

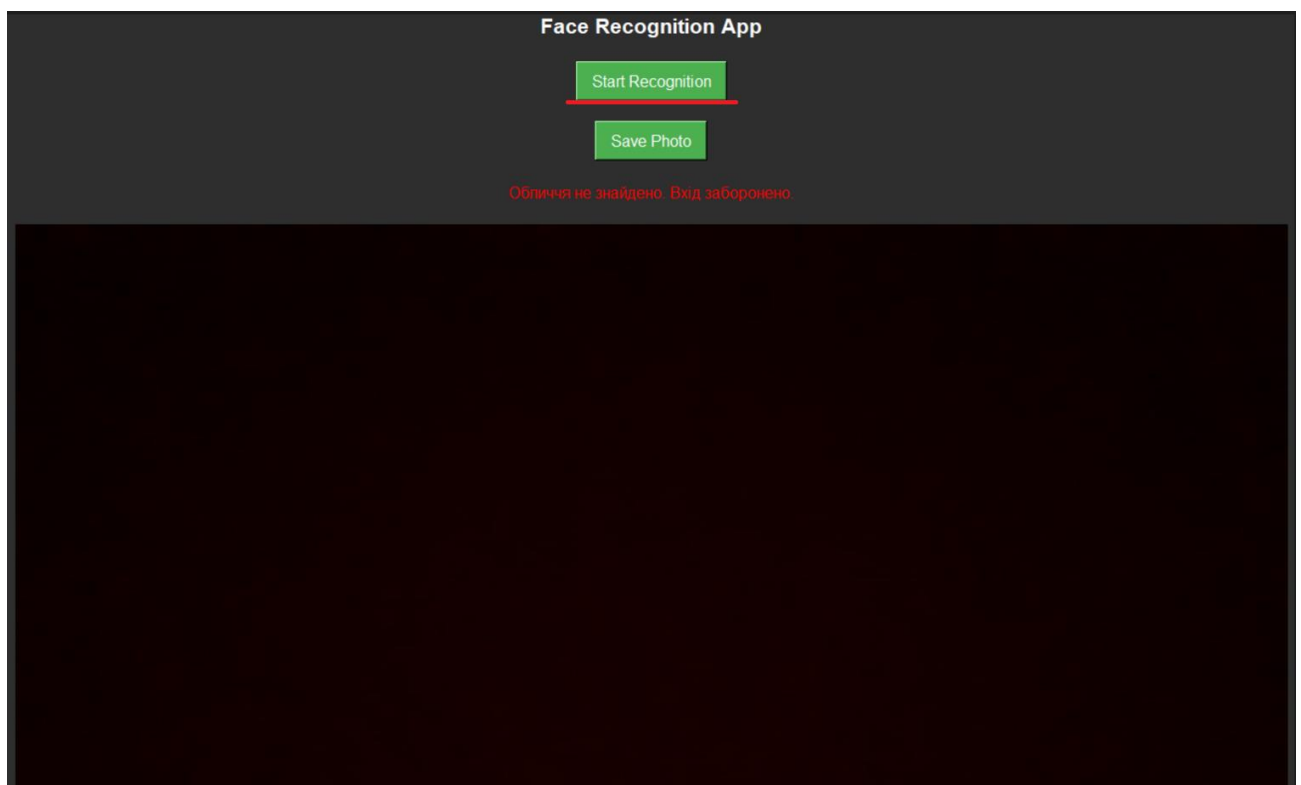


Рис.3.20. Реакція програми, якщо відсутнє обличчя у кадрі

На рисунку бачимо, як програма реагує на натискання кнопки Start Recognition. Після запуску розпізнавання, камера активується, але система не виявила жодних облич у кадрі. У такому випадку з’являється повідомлення: “Обличчя не знайдено. Вхід заборонено”:

1) кнопка Start Recognition була натиснута, що запустило процес обробки відеопотоку. Програма використовує функцію `start_recognition(self)` із файлу `combine.py`, де кожен кадр з камери аналізується з використанням методу `detectMultiScale()` з бібліотеки OpenCV. Цей метод намагається знайти обличчя на поточному кадрі, але, оскільки камера не зафіксувала жодного обличчя, програма переходить до наступного кроку;

2) у разі, коли обличчя не знайдено, програма відображає повідомлення через функцію `show_message(self, message, color)`, яка оновлює текст повідомлення і змінює його колір на червоний, сигналізуючи про помилку;

3) функція `start_recognition(self)` у файлі `combine.py` обробляє кожен кадр і в разі відсутності облич повертає виведення через метод `show_message`, яке повідомляє користувача про неможливість верифікації;

4) відеопотік залишається активним, але в області, де повинні відображатися захоплені кадри, обличчя не ідентифіковані, тому немає прямокутників навколо об'єктів на екрані;

5) вікно Employees Inform також не з'являється, оскільки система не змогла знайти обличчя і не може надати інформацію про співробітника.

Ця частина тестування демонструє, як програма реагує на ситуацію, коли обличчя не може бути виявлено на відеопотоці. Це важливий аспект тестування, який підтверджує, що програма коректно працює з негативними результатами і повідомляє користувача про відсутність доступу у випадку відсутності виявленого обличчя.

3. Третій етап: перевірка ідентифікації особи з бази даних. Результат можна побачити на рис.3.21.

На рисунку видно основний інтерфейс програми, у якому система розпізнає співробітника на підставі зображення з відеопотоку.

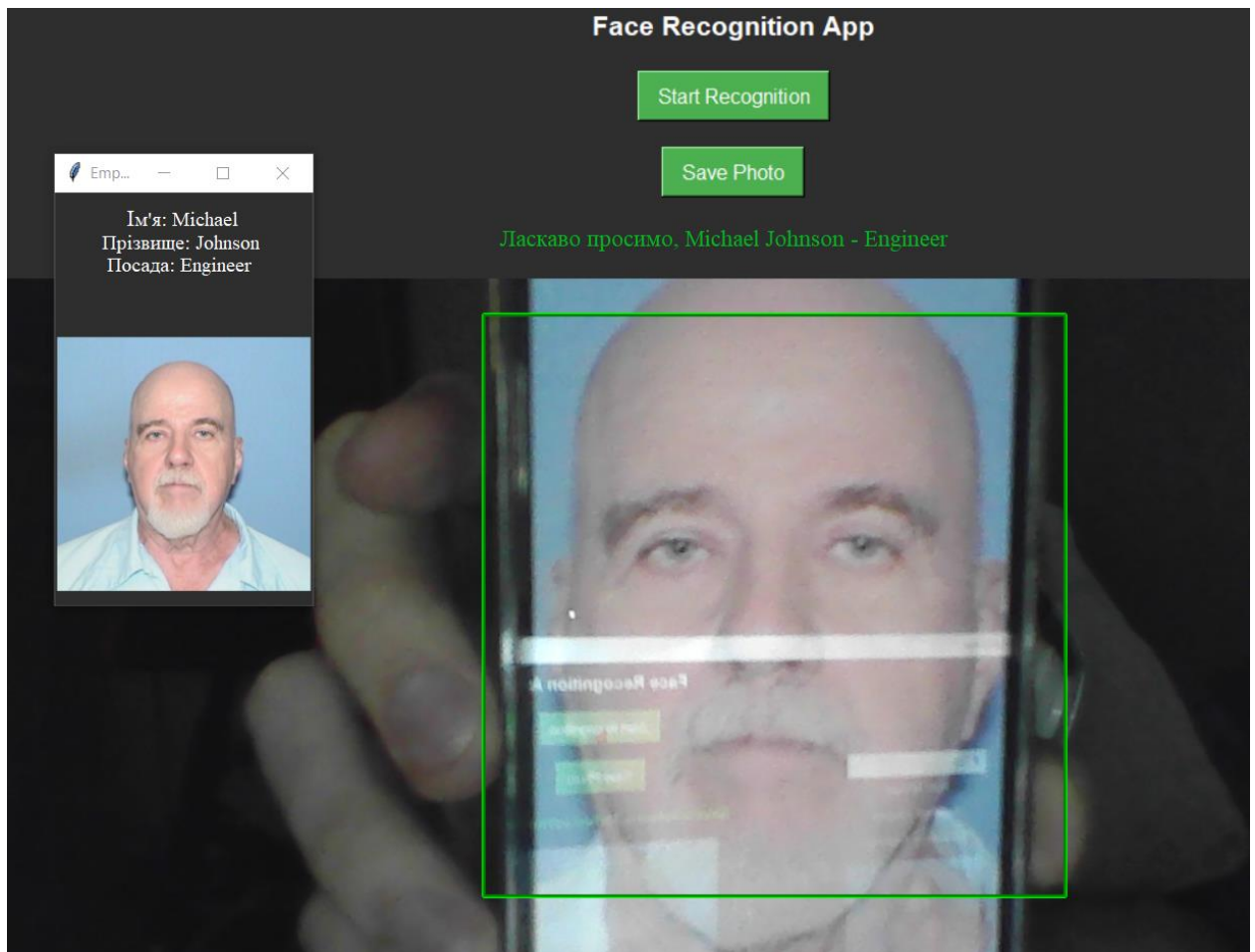


Рис.3.21. Результат розпізнання особи з бази даних

Тут можна виділити кілька важливих моментів:

1) процес розпізнавання обличчя починається після натискання кнопки Start Recognition, яка запускає захоплення відеопотоку з камери і передає кадри для обробки. Система виявляє обличчя на кожному кадрі за допомогою алгоритму детекції облич (реалізовано у файлі combine.py, метод start_recognition(self));

2) у кадрі з відеопотоку зелена рамка показує, що обличчя знайдено та оброблено. Це обличчя потрапило у програму, яка порівнює його з ембедінгами із бази даних, використовуючи модель розпізнавання облич, збережену в файлі your_model_weights.pth. Після цього, відповідно до отриманого результату, система порівнює ембедінги з базою даних у файлі YourDatabase, де метод get_most_similar_employee() знаходить найкращий збіг;

3) відображення інформації про співробітника зліва в інтерфейсі показано вікно з даними про знайдену особу – Ім'я: Michael, Прізвище: Johnson, Посада: Engineer. Це означає, що система знайшла у базі найближчий збіг і підтвердила, що на зображенні – інженер Michael Johnson. Цей результат отриманий після порівняння ембедінгів із базою;

4) привітання співробітника у центральній частині інтерфейсу є повідомлення “Ласкаво просимо, ...”, що свідчить про успішне розпізнавання. Це повідомлення оновлюється динамічно в залежності від результатів обробки кадрів.

Таким чином, на даному етапі бачимо успішну роботу системи, яка може ідентифікувати співробітників у реальному часі на основі відеопотоку з камери.

Після проведення трьох етапів тестування системи ідентифікації облич отримали важливі результати, які підтверджують функціональність та ефективність розробленої програми. Перший етап, що передбачав запуск програми без активного розпізнавання, дав змогу впевнитися у стабільності її роботи та коректності налаштувань. Другий етап, в якому система реагувала на відсутність обличчя, продемонстрував, що програма коректно обробляє такі ситуації і забезпечує безпеку доступу. Завершальний етап тестування, що включав ідентифікацію співробітників підтвердив, що система здатна точно розпізнавати обличчя та співвідносити їх з базою даних, забезпечуючи ефективність та надійність рішення.

Висновок до третього розділу

У цьому розділі представлені результати експериментальних досліджень системи ідентифікації осіб. Розробка та інтеграція прототипу системи включала в себе кілька важливих етапів, таких як підготовка даних, навчання моделі, створення архітектури нейронної мережі та тестування.

Вибір програмного середовища та бібліотек, таких як Python, OpenCV, TensorFlow, Keras та інших, забезпечив гнучкість та ефективність у розробці системи. Використання віртуального середовища (venv) дозволило ізолювати залежності проекту, що забезпечило стабільність роботи програми.

Після завершення етапу навчання моделі проведено три етапи тестування, які підтвердили функціональність та ефективність розробленої програми. Перший етап, що передбачав запуск програми без активного розпізнавання, продемонстрував стабільність роботи інтерфейсу. Другий етап, у якому програма реагувала на відсутність обличчя, показав, що система коректно обробляє ситуації без виявлених облич, надаючи користувачеві зрозуміле повідомлення про відсутність доступу. Завершальний етап тестування, що включав ідентифікацію співробітників, підтвердив, що система здатна точно розпізнавати обличчя та співвідносити їх із базою даних, забезпечуючи ефективність та надійність рішення.

На основі отриманих результатів можна стверджувати, що система має добрі перспективи для подальшого розвитку та вдосконалення, а реалізовані модулі та їх інтеграція забезпечують основу для розширення функціоналу системи у майбутньому.

РОЗДІЛ 4

АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ РОЗРОБЛЕНОЇ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ

4.1. Аналіз точності та продуктивності системи, впливальних на якість розпізнавання факторів.

У даному підрозділі проводиться аналіз точності та продуктивності розробленої системи нейромережевого розпізнавання облич.

Точність системи буде визначатися як співвідношення кількості правильних ідентифікацій до загальної кількості спроб. Це дозволяє оцінити ефективність алгоритму в умовах реального використання.

Продуктивність системи буде оцінюватися за допомогою вимірювання часу обробки зображень, а також швидкості розпізнавання в реальному часі. Це параметри є важливими для забезпечення зручності користування та інтерактивності системи.

Крім того, важливими факторами, які впливають на результат, є:

- якість зображень, що обробляються;
- умови освітлення, за яких здійснюється розпізнавання;
- положення обличчя в кадрі;
- варіації в зовнішньому вигляді об'єкта (наприклад, наявність окулярів або зміни зачіски).

Тому розпочнемо детальний розгляд конкретних показників точності та продуктивності системи на практиці, що надасть змогу глибше оцінити ефективність розробленої технології:

1. Оцінка точності системи: під час тестування системи на наборі даних із 20 працівників (рис.3.15) досягнуто таких результатів, як показано на рисунку 4.1.

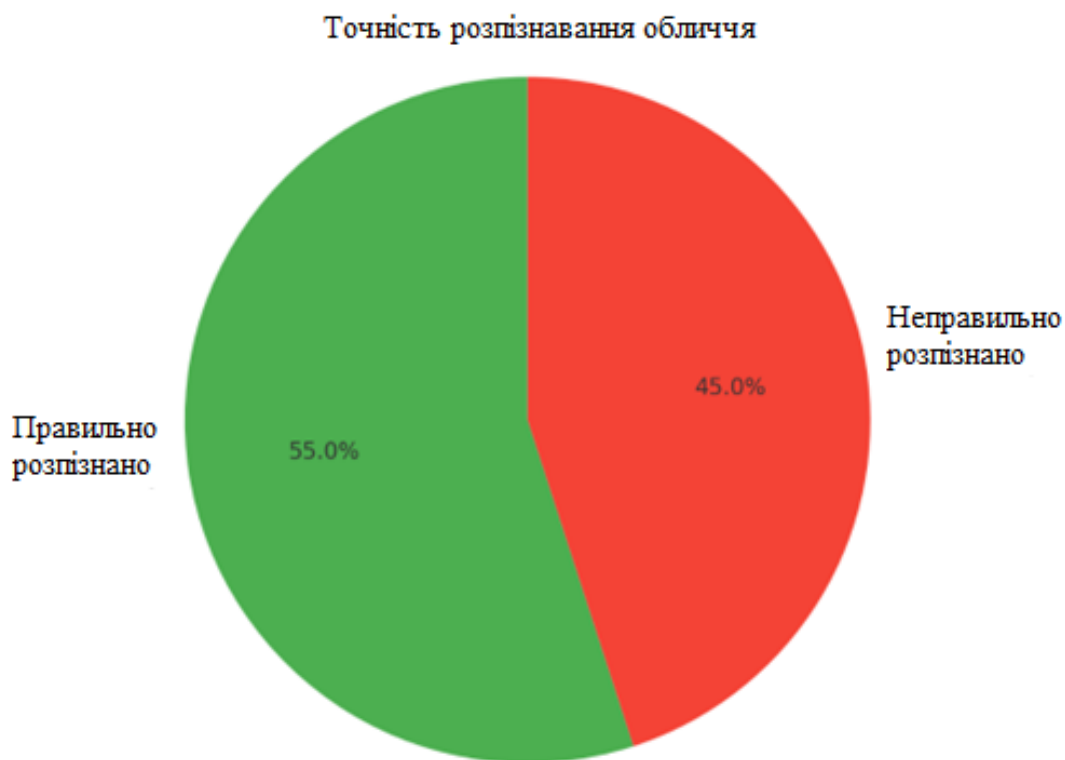


Рис.4.1. Графік оцінки точності системи

Система змогла правильно розпізнати 11 з 20 працівників, що становить 55% точності. Це відображено на графіку, де частина, позначена зеленим кольором вказує на коректно розпізнаних працівників, а червона частина – на некоректно розпізнаних. Відповідно, 45% працівників системи не змогла розпізнати або ж вони були визначені неправильно.

Основні причини такого рівня точності можуть включати:

- недостатня кількість навчальних прикладів для кожного працівника: кожна модель машинного навчання потребує великої кількості даних для тренування, і в цьому випадку може знадобитися збільшення кількості даних або різних варіацій облич (з різним освітленням, ракурсами тощо);
- особливості зображень: у реальних умовах на розпізнавання можуть впливати фактори, наприклад: якість зображень, різкі зміни освітлення, наявність окулярів або інших елементів, що можуть змінювати зовнішній вигляд обличчя;

- недосконалість моделі: модель потребує подальшого доопрацювання або використання більш потужних алгоритмів розпізнавання облич, які здатні враховувати тонкі відмінності між працівниками.

Таким чином, система вже продемонструвала непоганий рівень точності для базових умов, і її подальший розвиток сприятиме підвищенню надійності та продуктивності в реальних сценаріях використання.

2. Продуктивність системи: система розпізнавання облич працює в реальному часі, обробляючи відеопотік з камери з частотою приблизно 10-15 кадрів на секунду на процесорі Intel(R) Core(TM) i5-12450H. Це дозволяє виявляти та розпізнавати обличчя з великою швидкістю.

Під час тестування встановлено, що обробка кожного кадру, включно з процесами виявлення обличчя та класифікації, займала близько 80-130 мс. Завдяки цьому система може бути інтегрована в різні робочі середовища, де вимагається непогана швидкість реакції, як, наприклад, для контролю доступу або автоматизованого моніторингу.

Вплив продуктивності камери:

- розмір кадру: система працювала з відеопотоком розміром 1920x080 пікселів (рис.4.2), що забезпечило високу якість зображень для точного розпізнавання облич. Висока роздільна здатність дозволила системі з більшою точністю виявляти деталі обличчя, що є критичним для покращення ефективності розпізнавання, особливо у випадках, коли працівники перебувають на певній відстані від камери або частково перекриті іншими об'єктами;

- частота кадрів: камера була налаштована на 30 кадрів на секунду (рис.4.2), що дозволило безперебійно захоплювати та обробляти кожен кадр. Цей параметр є важливим для реальних сценаріїв використання, оскільки він забезпечує плавність зображення та мінімізує ймовірність пропусків моментів, коли обличчя може бути виявлено.

```

20 # Встановлюю роздільну здатність камери
21 self.video_capture.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
22 self.video_capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
23
24 self.video_capture.set(cv2.CAP_PROP_FPS, 30)

```

Рис.4.2. Встановлення розміру та частоти кадрів

Загалом, система показала достатню продуктивність для застосування в умовах реального часу, забезпечуючи швидке розпізнавання облич із мінімальною затримкою, що робить її придатною для широкого спектра застосувань.

3. Впливові фактори: ефективність та точність розпізнавання осіб за допомогою розробленої неймережевої системи можуть суттєво залежати від кількох факторів:

- Освітлення: належне освітлення є критично важливим для успішного розпізнавання облич. Недостатнє освітлення може призвести до появи тіней та затемнення, що ускладнює ідентифікацію. Яскраве світло або неправильне розташування джерел світла може викликати засвіти, які знижують контрастність зображення і втрачають важливі деталі, такі як контури обличчя;

- Якість відеокамери: використання відеокамери з високою роздільною здатністю дозволяє отримувати чіткіші та детальніші зображення, що суттєво полегшує процес розпізнавання. Рекомендується використовувати камери з роздільною здатністю не нижче 1080p для досягнення найкращих результатів;

- Кут огляду: кут, під яким обличчя розташоване відносно камери, також має велике значення. Ідеальний кут – фронтальний, оскільки бокові або верхні кути можуть зменшити точність розпізнавання;

- Динаміка кадру: швидкість руху об'єкта в кадрі може вплинути на якість розпізнавання. Якщо особа швидко рухається, зображення може бути розмитим, що ускладнює процес ідентифікації.

Ці фактори відіграють ключову роль у досягненні високої точності та ефективності системи розпізнавання облич, тому їх варто враховувати під час налаштування та експлуатації системи.

Даний аналіз точності показав, що цю систему можна надалі розвивати для покращення якості розпізнавання осіб. Однак, як зазначалося в підрозділі 3.1, обмеження апаратними можливостями не дає змоги розвивати систему далі.

Також під час аналізу виявлено, що за умов дотримання рекомендованих умов система може досягти високих показників розпізнавання, що підтверджує її потенціал для використання в різних сферах.

4.2. Рекомендації щодо покращення точності та надійності запропонованої системи.

Розроблена система розпізнавання облич демонструє хороші результати навіть за умов обмеженого апаратного забезпечення при навчанні, як зазначено у підрозділі 3.1. Однак апаратні обмеження вплинули на рішення обмежити кількість навчальних етапів до трьох, аби уникнути перенавчання та забезпечити стабільну роботу моделі. Але, незважаючи на досягнуті результати, існує потенціал для подальшого покращення точності та надійності системи.

Основні рекомендації щодо вдосконалення стосуються як поліпшення самої моделі, так і оптимізації процесів її роботи. Зважаючи на потребу збалансувати якість та продуктивність, можна виділити кілька ключових напрямків для розвитку:

1. Збільшення навчальної вибірки: додавання до набору навчальних даних нових зображень, зокрема з різними ракурсами та в умовах різного освітлення, допоможе моделі краще адаптуватися до змін облич користувачів.

Це дозволить підвищити точність розпізнавання в умовах реального використання;

2. Використання більш складних моделей: перехід на сучасніші архітектури нейронних мереж, такі як ResNet або FaceNet, відкриває можливість отримання більш точних результатів завдяки здатності цих моделей захоплювати більше важливих деталей;

3. Додаткове навчання моделі: постійне вдосконалення моделі за допомогою нових наборів даних дозволить системі адаптуватися до змін у робочому середовищі та підвищить точність розпізнавання обличч у специфічних умовах.

Щодо оптимізації системи, можна впровадити наступні заходи:

1. Оптимізація алгоритмів виявлення обличчя: використання спеціалізованих апаратних прискорювачів (GPU, TPU) може значно збільшити швидкість обробки відеопотоків без втрат точності. Це дасть змогу працювати на вищих частотах кадрів, що важливо для систем реального часу;

2. Багатопоточність: застосування багатопотокової обробки відеопотоків дозволить розподілити навантаження, прискорити процес розпізнавання та мінімізувати затримки в роботі системи;

3. Підвищення продуктивності на слабших пристроях: оптимізація системи дозволить використовувати її на менш потужному обладнанні без значних втрат у продуктивності, що розширить сферу її застосування.

Таким чином, хоча апаратні обмеження визначили певні рамки для розвитку системи, запропоновані рекомендації відкривають перспективи для подальшого покращення її точності, продуктивності та надійності.

Висновок до четвертого розділу

В рамках даного розділу проведено всебічне дослідження точності, продуктивності та факторів, що впливають на ефективність системи. В результаті аналізу виявлено, що система досягла базового рівня точності,

становлячи (55%) на даних із 20 працівників. Це свідчить про необхідність подальшого вдосконалення, особливо в умовах обмежених навчальних даних, що негативно впливають на результат.

Важливими аспектами, що впливають на точність і продуктивність, стали умови освітлення, якість відеокамери, розмір кадру та динаміка кадру. Визначено, що належне освітлення та якість камери є критично важливими для успішного розпізнавання облич.

Також, обґрунтовано можливості для вдосконалення системи. Основними напрямками стали збільшення навчальної вибірки, використання більш складних моделей та додаткове навчання. Окрім того, для оптимізації продуктивності системи запропоновано впровадження апаратних прискорювачів, багатопоточності та адаптації до слабших пристроїв.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Забезпечення охорони праці є важливим компонентом будь-якої професійної діяльності, спрямованим на створення безпечних та комфортних умов для працівників.

Охорона праці об'єднує сукупність правових, соціально-економічних, організаційних та технічних заходів, які мають на меті збереження здоров'я, життя і працездатності працівників у процесі виконання ними професійних обов'язків.

Головна мета охорони праці полягає у мінімізації або повному усуненні шкідливих і небезпечних факторів, які можуть впливати на працівників, зокрема на їх фізичне та психологічне здоров'я. Завдяки цьому вдається досягнути зменшенню травматизму та професійних захворювань, а також підвищується загальна ефективність роботи підприємства.

Згідно з Конституцією України, кожен громадянин має право на безпечні умови праці, які не завдають шкоди здоров'ю. Роботодавець зобов'язаний створити на робочому місці такі умови, які відповідали б стандартам безпеки, а також забезпечити захист працівників від потенційних ризиків і небезпек.

Окрім забезпечення фізичної безпеки працівників, охорона праці також спрямована на підвищення якості виробничого середовища, впровадження сучасних методів управління ризиками, що дозволяє не лише знизити кількість нещасних випадків, але й покращити загальний рівень життя працівників.

5.1. Шкідливі та небезпечні фактори, які впливають на розробника інформаційної системи.

Умови праці значно впливають на фізичний стан, продуктивність і загальний рівень здоров'я працівників. Усі фактори, що виникають у виробничому середовищі, можна поділити на шкідливі та небезпечні.

Небезпечні фактори – це фактори, що за певних умов можуть спричинити травму або різке погіршення стану здоров'я працівників. Наприклад, гарячі поверхні обладнання, струм високої напруги.

Шкідливі фактори – це ті, які поступово впливають на здоров'я працівників, призводячи до виникнення професійних захворювань. Наприклад, погіршення психоемоційного стану, тривала робота за комп'ютером та з поганим освітленням може спричинити перенапруження зору.

Шкідливі фактори можуть стати небезпечними у випадку їх значного посилення чи тривалого впливу.

Електротравми є одним із найпоширеніших небезпечних факторів, які можуть виникнути у виробничому середовищі. Сила струму, напруга, частота, тривалість впливу та стан організму визначають тяжкість ураження.

Основними видами електротравм є:

1. Локальні електротравми:

- опіки;
- електромарки (сліди струму на шкірі);
- механічні травми (трапляються через падіння або різкі рухи при впливі струму).

2. Загальні електротравми:

- ураження електричним струмом, яке викликає судоми, втрату свідомості чи навіть клінічну смерть.

Розрізняють чотири ступеня ураження електричним струмом:

- I ступінь: судомні болючі спазми в м'язах без втрати свідомості;
- II ступінь: судоми з втратою свідомості, але збереженням дихання та серцевої діяльності;
- III ступінь: сильні судоми, порушення дихання та серцевої діяльності;
- IV ступінь: клінічна смерть, повне припинення дихання та серцебиття.

Електричний струм може вплинути на організм тепловою, хімічною, механічною та біологічною дією. Наприклад, тепловий вплив спричиняє опіки, хімічний – зміну складу крові, механічний – розрив тканин, а біологічний – порушення роботи серця та нервової системи.

Робота за комп'ютером супроводжується впливом ряду фізичних факторів, які можуть призводити до погіршення здоров'я працівників. Ці фактори пов'язані як із технічними особливостями обладнання, так і з організацією робочого процесу.

Основними фізичними факторами є:

1. Електромагнітне випромінювання:
 - монітори створюють електромагнітне випромінювання низької частоти, яке тривалий час впливає на організм працівника. Воно може спричиняти головний біль, порушення сну та концентрації;
 - ультрафіолетове та інфрачервоне випромінювання, хоча й мінімальне у сучасних пристроях, залишається потенційно шкідливим за умов тривалої роботи.
2. Підвищений рівень пульсації світлового потоку:
 - некоректно налаштовані монітори або погане освітлення робочого місця можуть спричиняти втому зору та головні болі.
3. Запиленість повітря:

- погана вентиляція приміщення, де працюють працівники, призводить до накопичення пилу, який може осідати на очах, дихальних шляхах або шкірі, викликаючи алергію або подразнення.

4. Шум і вібрація:

- деякі пристрої, наприклад, принтери або сканери можуть створювати підвищений рівень шуму та вібрацій, що негативно впливає на нервову систему працівників.

5. Нерівномірний розподіл яскравості у полі зору:

- контраст між яскравістю екрана та освітленням приміщення викликає перенапруження і швидке стомлення.

Заходами для мінімізації ризиків є:

- використання сучасних моніторів із низьким рівнем випромінювання;
- забезпечення рівномірного та достатнього освітлення робочого місця;
- встановлення шумопоглинаючих матеріалів у приміщеннях із високим рівнем шуму;
- регулярно прибирати приміщення для зменшення запиленості.

Також, робота за комп'ютером передбачає тривале статичне навантаження, інтенсивну розумову діяльність та концентрацію уваги, що може призводити до психоемоційного виснаження працівників. Ці фактори не лише впливають на працездатність, але й створюють ризик розвитку професійних захворювань.

Основні психофізіологічні фактори поділяються на:

1. Навантаження на пам'ять та увагу:

- постійна робота з великим обсягом інформації та необхідність її аналізу створюють високе навантаження на когнітивні функції;
- умови багатозадачності можуть викликати стрес і зниження продуктивності.

2. Тривале статичне напруження:

- довготривале сидіння у фіксованій позі сприяє розвитку захворювань опорно-рухового апарату (викривлення хребта, остеохондрозу);
- неправильна організація робочого місця посилює ризик цих проблем.

3. Напруження зору:

- тривала роботи за монітором призводить до синдрому комп'ютерного зору, який характеризується втомою, сухістю очей, розмитим баченням і головними болями.

4. Монотонність роботи:

- виконання однотипних завдань протягом тривалого часу може викликати емоційне вигорання, втрату мотивації та загальну втому.

5. Нераціональна організація робочого процесу:

- відсутність перерв для відпочинку, надмірне навантаження або нечітка постановка завдань посилюють негативний вплив психофізіологічних факторів.

Шляхи мінімізації цих факторів є:

- організація регулярних перерв для коротких фізичних вправ і відпочинку очей;
- використання ергономічних меблів для зниження статичного напруження;
- впровадження автоматизованих інструментів для обробки великих обсягів даних з метою зменшення навантаження на пам'ять та увагу;
- забезпечення різноманітності завдань, щоб уникнути монотонності роботи.

Ефективна організація робочого місця розробника програмного забезпечення є важливим елементом створення безпечних та комфортних умов праці. Правильно підібрана конфігурація обладнання та раціональне розташування компонентів дозволяють зменшити негативний вплив виробничих факторів та підвищити продуктивність.

Основними компонентами робочого місця є:

1. Пристрій відображення інформації (монітор):

- монітор повинен мати високу роздільну здатність, антивідблискове покриття та можливість регулювання яскравості для зниження навантаження на зір;

- відстань від очей до екрана повинна становити 50–70 см, а верхній край монітора — знаходитися на рівні очей або трохи нижче.

2. Персональний комп'ютер:

- персональний комп'ютер з базовим набором пристроїв введення-виведення, таких як клавіатура, миша та периферійні пристрої;

- живлення обладнання повинно здійснюватися від надійного джерела електроживлення з заземленням, що знижує ризик ураження електричним струмом.

3. Меблі для роботи:

- ергономічний стілець із регульованою висотою, підтримкою поперекового відділу спини та можливістю змінювати нахил спинки;

- робочий стіл достатнього розміру для розташування всього необхідного обладнання та матеріалів.

4. Освітлення:

- робоче місце повинно мати достатній рівень освітлення, зокрема комбіноване природне та штучне світло. Необхідно уникати прямих відблисків на моніторі.

5. Засоби захисту:

- використання джерел безперебійного живлення (UPS) для захисту обладнання від перепадів напруги;

- антистатичні килимки для зменшення впливу статичної електрики.

Рекомендації щодо організації робочого процесу:

- забезпечити регулярні перерви для відпочинку працівників, включаючи вправи для очей та короткі фізичні розминки;

- впроваджувати автоматизацію рутинних завдань, щоб зменшити навантаження на працівників і підвищити ефективність роботи;
- регулярно перевіряти стан обладнання та вживати заходів для профілактики технічних несправностей.

Робота з інформаційними системами, зокрема системами ідентифікації осіб, вимагає забезпечення високого рівня інформаційної безпеки. Недотримання вимог захисту може призвести до втрати даних, несанкціонованого доступу до конфіденційної інформації або навіть до порушення працездатності системи.

Основні загрози інформаційній безпеці:

1. Порушення конфіденційності:
 - несанкціонований доступ до конфіденційних даних працівників або користувачів системи;
 - розголошення або витік персональних даних через хакерські атаки чи недостатній рівень захисту.
2. Порушення цілісності даних:
 - спотворення інформації внаслідок програмних помилок або зловмисного втручання;
 - втрата важливих даних через дії шкідливого програмного забезпечення.
3. Недоступність системи:
 - атаки типу «відмова в обслуговуванні» (DoS), які блокують роботу інформаційної системи;
 - технічні несправності обладнання, що обмежують доступ до ресурсів системи.

Джерелами загроз є:

- Техногенні: віруси, троянські програми, фішингові атаки, використання вразливостей у програмному забезпеченні;
- Людський фактор: помилки співробітників, недбалість, відсутність належної підготовки у сфері кібербезпеки;

- Природні явища: стихійні лиха (пожежі, затоплення), які можуть пошкодити обладнання чи порушити функціонування системи.

Заходи для забезпечення інформаційної безпеки:

1. Технічні заходи:

- використання сучасного програмного забезпечення для захисту від вірусів і шкідливих програм;

- впровадження систем резервного копіювання даних для мінімізації втрат у разі технічних несправностей.

2. Організаційні заходи:

- контроль доступу до інформаційних ресурсів за допомогою авторизації та багаторівневого захисту;

- регулярне навчання співробітників основам інформаційної безпеки.

3. Фізичний захист обладнання:

- захист серверів та комп'ютерної техніки від фізичних пошкоджень і стихійних лих;

- організація доступу до серверних кімнат лише для уповноважених осіб.

Забезпечення інформаційної безпеки є невід'ємною частиною роботи з інформаційними системами, особливо у сфері ідентифікації осіб, де точність і захист даних мають вирішальне значення.

Ефективний захист інформації передбачає застосування комплексного підходу, що включає технічні, організаційні та фізичні заходи. Кожен із них спрямований на забезпечення конфіденційності, цілісності та доступності даних.

Основними засобами захисту є:

1. Технічні засоби:

- міжмережеві екрани (фаєрволи): блокують несанкціонований доступ до мережевих ресурсів;

- антивірусне програмне забезпечення: забезпечує виявлення та видалення шкідливих програм;
- системи шифрування: захищають дані від несанкціонованого доступу під час передачі або зберігання;
- системи резервного копіювання: дозволяють відновити дані у разі їх втрати або пошкодження.

2. Організаційні заходи:

- встановлення правил доступу до інформації, зокрема багаторівневої аутентифікації;
- Розробка та впровадження політик інформаційної безпеки;
- навчання співробітників основам кібербезпеки, включаючи протидію фішинговим атакам і правильному поводженню з конфіденційними даними.

3. Фізичні заходи:

- обладнання серверних кімнат засобами контролю доступу (замки, камери спостереження);
- використання сейфів або захищених контейнерів для зберігання носіїв даних;
- організація захисту обладнання від впливу зовнішніх чинників, таких як пил, волога або температура.

4. Моніторинг і аудит:

- регулярний перегляд журналів подій і аналіз підозрілих дій;
- впровадження системи моніторингу мережі для виявлення можливих загроз у режимі реального часу;
- проведення аудитів інформаційної безпеки для перевірки відповідності системи встановленим стандартам.

Заходи, які мінімізують ризики:

- впровадження системи оновлення програмного забезпечення для усунення вразливостей;

- використання сегментації мережі для обмеження можливостей зловмисників поширюватися системою;
- створення інструкцій з дій у разі виникнення інцидентів безпеки (інцидент-менеджмент).

Забезпечення комплексного захисту даних є критично важливим для стабільної роботи будь-якої інформаційної системи, особливо тих, що використовуються у бізнес-процесах.

Забезпечення безпеки інформаційних систем є багатогранним завданням, яке вимагає інтегрованого підходу. У процесі аналізу загроз і розробки захисних заходів виявлено, що ефективність системи безпеки залежить не тільки від технічних засобів, а й від організаційної структури та рівня обізнаності персоналу.

Рекомендації:

1. Постійне оновлення програмного забезпечення:
 - використання актуальних версій програмного забезпечення для усунення вразливостей;
 - регулярне оновлення антивірусних баз і засобів захисту мережі.
2. Розширення системи моніторингу:
 - впровадження інструментів аналізу мережевої активності для своєчасного виявлення аномалій;
 - застосування систем автоматичного оповіщення про потенційні загрози.
3. Навчання персоналу:
 - проведення тренінгів із кібербезпеки для всіх співробітників;
 - ознайомлення з основними методами запобігання фішинговим атакам та інших форм шахрайства.
4. Впровадження резервного копіювання:
 - розробка плану відновлення даних після інцидентів;
 - використання віддалених серверів або хмарних технологій для зберігання резервних копій.

5. Фізична ізоляція критичних систем:

- захист серверних приміщень та інших важливих зон від несанкціонованого доступу;
- використання датчиків температури, вологості та інших інструментів моніторингу умов експлуатації.

Виконання цих рекомендацій дозволить суттєво підвищити рівень захисту інформаційних систем, забезпечивши їхню стабільну роботу та збереження конфіденційності, цілісності й доступності даних.

5.2. Засоби зменшення впливу шкідливих та небезпечних факторів, які діють на розробника інформаційної системи.

Зменшення впливу небезпечних і шкідливих факторів на робочому місці є одним із ключових завдань охорони праці. Для досягнення безпеки працівників застосовуються як колективні, так і індивідуальні засоби захисту, що спрямовані на створення комфортних умов праці та мінімізацію ризиків травмування або захворювань.

Засоби захисту класифікують за такими типам:

1. Колективні засоби захисту. Ці заходи забезпечують загальну безпеку робочого середовища і спрямовані на зниження ризиків впливу зовнішніх факторів на всіх працівників:

- контроль якості повітря на робочому місці (вентиляція, очищення та кондиціонування повітря);
- мінімізація рівня шуму й вібрації за допомогою спеціальних огорож і звукоізоляційних матеріалів;
- захист від електромагнітних випромінювань і іонізуючих полів за допомогою захисних бар'єрів, екранів і сигналізацій;
- нормування освітлення для зменшення навантаження на зір, використання антивідблискових пристроїв.

2. Індивідуальні засоби захисту. Забезпечують захист конкретного працівника і використовуються відповідно до специфіки виконуваних робіт. Основними видами є:

- засоби захисту органів дихання (маски, респіратори);
- захисний одяг (комбінезони, куртки, рукавиці);
- окуляри або щитки для захисту очей від пилу, випромінювання чи механічних частинок;
- засоби захисту слуху.

Колективні засоби захисту створюють безпечні умови для всіх працівників, забезпечуючи мінімізацію впливу небезпечних і шкідливих факторів. Їх ефективність досягається через системне застосування технічних, організаційних і технологічних заходів.

Основними напрямками колективного захисту є:

1. Нормування повітряного середовища:
 - використання систем вентиляції для забезпечення постійного оновлення повітря в приміщеннях;
 - установлення кондиціонерів для підтримання оптимальної температури та вологості;
 - фільтрація повітря для усунення пилу, шкідливих газів і забруднень.
2. Захист від шуму та вібрацій:
 - звукоізоляційні екрани для зменшення рівня шуму, що виходить від обладнання;
 - віброізоляційні матеріали, які встановлюються на джерела вібрації або робочі поверхні.
3. Нормування освітлення:
 - використання джерел природного та штучного освітлення з урахуванням специфіки виконуваних завдань;
 - використання ламп із регульованою яскравістю.
4. Захист від електромагнітного та іонізуючого випромінювання:

- застосування захисних екранів, які блокують електромагнітне випромінювання від моніторів і комп'ютерного обладнання;

- використання автоматичних систем контролю рівня випромінювання.

5. Запобігання ураженню електричним струмом:

- встановлення захисного заземлення на обладнанні;
- автоматичні вимикачі, які відключають живлення у разі короткого замикання;

- дистанційне керування обладнанням, щоб уникнути контакту з небезпечними зонами.

6. Температурний контроль:

- утеплення робочих зон на низьких температурах;
- автоматичні системи регулювання температури для запобігання перегріву або переохолодженню працівників;

- використання теплових екранів для ізоляції нагрітих поверхонь.

Коллективні засоби захисту мають ключову роль у створенні безпечних і комфортних умов праці, забезпечуючи масовий захист працівників від негативного впливу виробничих факторів.

Індивідуальні засоби захисту (ІЗЗ) розроблені для мінімізації негативного впливу шкідливих і небезпечних факторів на конкретного працівника. Їх використання дозволяє забезпечити додатковий рівень безпеки при виконанні специфічних завдань або роботи в особливо небезпечних умовах.

Індивідуальні засоби захисту класифікують за такими типами:

1. Захист очей і обличчя:

- захисні окуляри: використовуються для запобігання травми очей від механічних часток або випромінювання;

- щитки та маски: забезпечують повний захист обличчя від термічних чи хімічних впливів.

2. Захист шкіри та рук:

- рукавиці та рукавички: виготовлені з матеріалів, стійких до хімічних, термічних чи механічних впливів;
- дерматологічні засоби захисту: спеціальні креми, паста або мазі для зменшення подразнень і запобігання впливу агресивних речовин.

3. Захист від падінь:

- запобіжні пояси: застосовуються при роботі на висоті для запобігання падіння;
- діелектричні килимки: використовуються для ізоляції від електричного струму.

4. Захист тіла:

- ізолюючі костюми: пневмокостюми або скафандри для роботи в умовах підвищеної небезпеки;
- спеціальний одяг: комбінезони, куртки, брюки, які забезпечують захист від фізичних або хімічних впливів;
- теплозахисний одяг: використовується для роботи при високих або низьких температурах.

5. Захист органів слуху:

- протишумові навушники: знижують рівень шуму, що впливає на слуховий апарат;
- беруші: ефективні для роботи в середовищі з підвищеним рівнем шуму.

Застосування ІЗЗ дозволяє значно зменшити ризик травм або захворювань, викликаних шкідливими чинниками. Вони є обов'язковим елементом захисту на виробництві, де наявні високі ризики, і часто доповнюють колективні засоби захисту, забезпечуючи максимальну безпеку працівників.

Комп'ютерне обладнання є невід'ємною частиною роботи розробника інформаційних систем. Відповідна організація робочого місця та дотримання правил безпеки дозволяють мінімізувати негативний вплив шкідливих факторів та забезпечити комфортні умови праці.

До рекомендацій організації робочого місця відносяться:

1. Правильне розташування обладнання:
 - монітор має бути розташований під прямим кутом до очей користувача, трохи нижче рівня погляду;
 - системний блок, принтер та інші пристрої потрібно розташувати так, щоб забезпечити зручний доступ до них, уникаючи зайвих нахилів чи рухів.
2. Регулювання освітлення:
 - забезпечити рівномірне освітлення робочого місця, уникати яскравих відблисків на моніторі;
 - використовувати світлозахисні пристрої або антивідблискове покриття на екрані.
3. Вентиляція приміщення:
 - регулярно провітрювати приміщення для підтримання належного рівня кисню;
 - використовувати кондиціонери або системи очищення повітря для створення комфортного мікроклімату.
4. Ергономіка робочого місця:
 - стілець повинен мати регулювання висоти;
 - руки мають бути розташовані під прямим кутом при використанні клавіатури, зап'ястя на рівні поверхні.

Основними правилами безпеки є:

1. Перевірка обладнання перед роботою:
 - переконатися у справності кабелів, розеток та заземлення;
 - уникати перевантаження електромережі, використовуючи мережеві фільтри та стабілізатори напруги.
2. Порядок увімкнення та вимкнення пристроїв:
 - увімкнення обладнання має виконуватись у наступному порядку: монітор, системний блок, периферійні пристрої;

- вимикання здійснюється у зворотному порядку, при цьому всі кабелі слід від'єднувати лише після повного вимкнення.

3. Налаштування монітора:

- встановити оптимальну яскравість і контрастність зображення, щоб зменшити навантаження на очі;

- зображення не повинно бути надто яскравим або тьманим.

4. Перерви в роботі:

- Робити короткі перерви кожні 45-60 хвилин для розминки очей і запобігання статичному навантаженню;

- виконувати вправи для розслаблення м'язів спини та ший.

Дотримання цих правил не лише підвищує рівень безпеки працівників, але й сприяє збереженню здоров'я та підвищенню продуктивності роботи.

5.3. Охорона навколишнього природного середовища.

Забезпечення екологічної безпеки та захисту навколишнього природного середовища є важливим аспектом діяльності будь-якого підприємства, включаючи ті, що впроваджують інноваційні технології, такі як нейромережеві системи. З огляду на специфіку розробки та експлуатації інформаційних систем, ключовими завданнями стають мінімізація шкідливого впливу на довкілля під час роботи комп'ютерного обладнання, оптимізація енергоспоживання та дотримання санітарно-гігієнічних норм.

Інтеграція штучного інтелекту у виробничі процеси відкриває широкі можливості для автоматизації моніторингу екологічних показників, що дозволяє зменшити ризики негативного впливу на навколишнє середовище.

Для запобігання негативному впливу впровадження нейромережевих систем і їх експлуатації на довкілля розробляються спеціальні заходи, що спрямовані на раціональне використання ресурсів, зниження рівня забруднення та забезпечення екологічної безпеки.

Основними напрямками екологічних заходів є:

1. Розробка нормативно-правової бази:
 - розробка внутрішніх стандартів екологічної безпеки для підприємства, яке впроваджує і експлуатує системи ідентифікації осіб;
 - включення екологічних вимог до технічних регламентів під час розробки інформаційних систем.
2. Контроль за енергоспоживанням:
 - оптимізація енергоспоживання серверного обладнання шляхом використання енергоефективних компонентів;
 - впровадження автоматизованих систем керування живленням, які дозволяють зменшити споживання електроенергії під час простоїв обладнання.
3. Моніторинг забруднень:
 - постійний контроль рівня електромагнітного випромінювання від комп'ютерного обладнання;
 - аналіз повітряного середовища в приміщеннях для виявлення та усунення надлишкової концентрації озону, пилу або інших шкідливих речовин.
4. Утилізація електронних відходів:
 - організація безпечного збору та утилізації застарілого або несправного обладнання відповідно до екологічних стандартів;
 - використання послуг сертифікованих компаній для переробки електронних відходів.
5. Оптимізація вентиляції та кондиціонування:
 - використання сучасних фільтраційних систем для очищення повітря в приміщеннях;
 - обладнання приміщень системами рекуперації тепла, що дозволяє знизити енергоспоживання.
6. Організація робочих місць:
 - установка обладнання, що знижує рівень електромагнітного випромінювання;

- оснащення робочих місць системами автоматичного моніторингу якості повітря.

Екологічний паспорт підприємства є офіційним документом, що відображає основні екологічні показники діяльності та заходи, які здійснюються для зменшення впливу на довкілля. У випадку впровадження нейромережових систем, таких як системи ідентифікації осіб, екологічний паспорт є важливим інструментом забезпечення відповідності екологічним стандартам.

Основні розділи екологічного паспорта:

1. Схеми очищення повітря та води;
2. Характеристика технологій;
3. Дані про відходи;
4. Інформація про шкідливі викиди;
5. Загальна інформація про підприємство.

Екологічний паспорт допомагає контролювати вплив підприємства на довкілля, забезпечує прозорість у відносинах із контролюючими органами та сприяє впровадженню передових технологій для охорони природи.

З метою зменшення впливу діяльності підприємства, яке використовує нейромережові системи ідентифікації осіб, на навколишнє середовище впроваджуються комплексні заходи. Вони охоплюють організаційні, технічні та технологічні рішення, спрямовані на мінімізацію ризиків забруднення та забезпечення сталого розвитку.

Основними заходами є:

1. Оформлення необхідних дозвільних документів:
 - проведення інвентаризації джерел шкідливих викидів у повітря;
 - отримання дозволів на викиди шкідливих речовин відповідно до чинного законодавства;
 - підготовка паспортів та реєстраційних карток для кожного виду промислових відходів;

- інвентаризація всіх відходів, утворених під час експлуатації систем ідентифікації осіб.

2. Взаємодія з державними та екологічними організаціями:

- співпраця з Міністерством екології та природних ресурсів України для впровадження передових природоохоронних стандартів;

- регулярна звітність перед державними органами щодо екологічної ситуації на підприємстві;

- взаємодія з місцевими органами влади та громадськими організаціями для покращення екологічного стану регіону.

3. Контроль забруднень:

- моніторинг рівня викидів шкідливих речовин у повітря, включаючи електромагнітне випромінювання від серверного обладнання;

- контроль за станом ґрунту та води на прилеглих територіях;

- встановлення газоочисного обладнання для зменшення шкідливого впливу на атмосферу.

4. Зниження рівня енергоспоживання:

- використання обладнання з високою енергоефективністю;

- впровадження «зелених» джерел енергії, таких як сонячні батареї або вітряні генератори.

5. Організація безпечного зберігання відходів:

- обладнання спеціальних майданчиків для сортування та зберігання відходів;

- використання контейнерів із підвищеною герметичністю для небезпечних відходів;

- організація системи регулярного вивезення відходів для їх утилізації.

6. Покращення умов на робочих місцях:

- оснащення робочих зон газосигналізаторами для своєчасного виявлення шкідливих речовин;

- дотримання санітарно-гігієнічних норм для забезпечення комфортного мікроклімату;

7. Підвищення екологічної обізнаності персоналу:

- проведення навчань із питань охорони навколишнього середовища;
- заохочення працівників до участі у програмах сталого розвитку;
- впровадження системи екологічних ініціатив, спрямованих на збереження природних ресурсів.

Таким чином, системний підхід до мінімізації екологічних ризиків дозволяє підприємствам, які використовують сучасні інформаційні технології, досягти балансу між ефективністю діяльності та збереженням навколишнього середовища.

Висновок до п'ятого розділу

У рамках даного розділу проведено аналіз умов праці розробників інформаційних систем, визначено шкідливі та небезпечні фактори, а також досліджено заходи для зменшення їх впливу. Виявлено, що основними ризиками є електромагнітне випромінювання, статична електрика, вплив шуму та вібрацій, а також порушення ергономіки робочих місць.

Оцінено ефективність впровадження колективних і індивідуальних засобів захисту. Зокрема, встановлено, що використання сучасних систем вентиляції, захисних екранів, антивідблискових покриттів та ергономічного обладнання значно покращує умови праці.

Особливу увагу приділено питанням охорони навколишнього середовища. Проведений аналіз показав, що оптимізація енергоспоживання, моніторинг викидів шкідливих речовин, утилізація електронних відходів та впровадження екологічного паспорту підприємства є ключовими заходами для мінімізації екологічних ризиків.

ВИСНОВКИ

У даній магістерській роботі розроблено нейромережеву систему ідентифікації особи на підприємстві. Проведено аналіз сучасних підходів до створення систем розпізнавання облич, огляд основних типів нейронних мереж та обрано оптимальну архітектуру для вирішення завдання ідентифікації особи. Розглянуто різні методи й алгоритми, що використовуються у нейронних мережах для досягнення високої точності розпізнавання, та визначено ключові фактори, що впливають на ефективність таких систем.

Результатом роботи стало створення функціональної нейромережевої системи, яка інтегрує інтерфейс для зручної взаємодії з користувачем, базу даних співробітників та модуль обробки зображень. Система розроблена на основі мови програмування Python із використанням бібліотек OpenCV та PyTorch, що дозволило досягти високої продуктивності в умовах обмеженого апаратного забезпечення. Завдяки інтеграції GUI користувач має можливість запускати розпізнавання, зберігати зображення та отримувати результати ідентифікації у зручному форматі.

Під час тестування система показала точність на рівні 55% при розпізнаванні облич, обробляючи відеопотік у реальному часі з частотою 10-15 кадрів на секунду. Також, підтверджено стабільну роботу програми та її здатність виконувати завдання розпізнавання з мінімальними затримками, що робить її ефективним інструментом для використання у сфері контролю доступу на підприємствах.

У роботі також розглянуто питання охорони праці. Проведено аналіз потенційних небезпечних факторів, що можуть впливати на людину під час роботи із системою, та розроблено рекомендації для забезпечення безпеки у процесі експлуатації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python. Налаштування [Електронний ресурс]. – Режим доступу: <https://www.python.org>
2. Tkinter – Python interface to Tcl/Tk [Електронний ресурс]. – Режим доступу: <https://docs.python.org/3/library/tkinter.html>
3. Visual Studio Code. Налаштування [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/docs/supporting/faq>
4. Python. Вивчіть найважливішу мову для науки про дані [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/learn/python>
5. Kaggle. Переходь на рівень із найбільшою спільнотою AI & ML [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com>
6. Лутц М. Вивчаємо Python : том 2, 5-те видання / М. Лутц. – Київ : Наукова думка, 2020. – 722 с. – Режим доступу: https://library.kre.dp.ua/Books/2-4%20kurs/Мови%20програмування/Python/Izuchaem_Python_Tom_2_5-e_izdanie_Mark_Lutts.pdf
7. Venv – створення віртуальних середовищ [Електронний ресурс]. – Режим доступу: <https://docs.python.org/3/library/venv.html>
8. Алгоритм оптимізації на основі глибокого навчання для аутентифікації особистості [Електронний ресурс]. – Режим доступу: <https://onlinelibrary.wiley.com/doi/full/10.1155/2022/9662817>
9. Нейронні мережі в перевірці ідентифікації: як ШІ оптимізує процес [Електронний ресурс]. – Режим доступу: <https://regulaforensics.com/blog/neural-networks/>
10. Дауні А. Основи Python. Навчіться думати як програміст / А. Дауні. – Київ : МАНН, 2021. – 306 с. – Режим доступу: https://library.kre.dp.ua/Books/2-4%20kurs/Мови%20програмування/Python/Allen_B_Dauni_Osnovy_Python_Nauchites_dumat_kak_programmist_Mann.pdf

11. Метиз Е. Вивчаємо Python: програмування ігор, візуалізація даних, веб-програми / Е. Метиз. – Київ : Літера, 2020. – 512 с. – Режим доступу: <https://l-p.ua/wp-content/uploads/2021/10/metiz.-izuchaem-python.pdf>

12. Терейковський І. А., Бушуєв Д. А., Терейковська Л. О. Штучні нейронні мережі: базові положення / І. А. Терейковський, Д. А. Бушуєв, Л. О. Терейковська. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 123 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/9fee52b6-83fc-4e99-8541-c2767f634c7c/content>

13. Лекція 3. Штучні нейронні мережі [Електронний ресурс]. – Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/176771/mod_resource/content/1/III_КБ_Л-3_Нейромережи.pdf

14. Яковенко А. В. Основи програмування. Python : частина 1 / А. В. Яковенко. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 185 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/dbbe8ff5-11d7-4a92-918c-d1445c3d20a7/content>

15. OpenCV – найбільша бібліотека комп'ютерного зору [Електронний ресурс]. – Режим доступу: <https://opencv.org>

16. Нейробіологічні аспекти розпізнавання обличчя [Електронний ресурс]. – Режим доступу: <https://www.frontiersin.org/journals/behavioral-neuroscience/articles/10.3389/fnbeh.2018.00195/full>

17. Принцип побудови нейронних мереж [Електронний ресурс]. – Режим доступу: <https://itmaster.biz.ua/programming/vision/neural-networks-principles.html>