

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ ПОШУКУ І РОЗПІЗНАВАННЯ
ОСОБИ У ВІДЕОЗОБРАЖЕННІ**

Виконав: студент 2 курсу, групи 1КІ-23м
спеціальності 123 — «Комп'ютерна інженерія»

Кардаш В. О. Кардаш В. О.

Керівник: д.т.н., проф. каф. ОТ

Мартинюк Т. Б. Мартинюк Т. Б.

«12» 12 2024 р.

Опонент: к.т.н., доц. каф. ПЗ

Черноволик Г. О. Черноволик Г. О.

«12» 12 2024 р.

Допущено до захисту

Завідувач кафедри ОТ

Азаров О.Д. д.т.н., проф. Азаров О.Д.

"12" 12 2024 р.

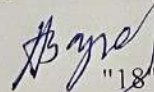
ВНТУ 2024

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Галузь знань — Інформаційні технології
Освітній рівень — магістр
Спеціальність — 123 Комп'ютерна інженерія
Освітня програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки



О. Д. Азаров

"18" вересня 2024 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту **КАРДАШУ Віталію Олександровичу**

1 Тема роботи «Програмний комплекс для пошуку і розпізнавання особи у відеозображенні» керівник роботи Мартинюк Тетяна Борисівна д.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу від «17» вересня 2024 року № 310.

2 Строк подання студентом роботи: 12.12.24.

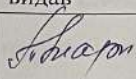
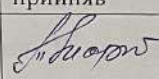
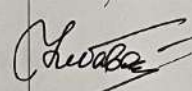
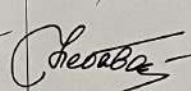
3 Вихідні дані до роботи: технічний опис сучасних методів та засобів розпізнавання особи у відеозображенні, середовище розробки ПЗ Visual Studio.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз методів та засобів розпізнавання особи у відеозображенні, розробка технології виявлення і розпізнавання особи у відеозображенні, розробка програми виявлення і розпізнавання особи у відеозображенні, перевірка працездатності програмного продукту, розрахунок економічної доцільності створення програми виявлення і розпізнавання особи у відеозображенні.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): методи розпізнавання обличчя людини, типи нейронних мереж, структура програми розпізнавання особи, інтелектуальні методи розпізнавання.

6 Консультанти розділів проекту (роботи)

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис,	дата
		завдання видав	завдання прийняв
1-4	д. т. н., професор Мартинюк Тетяна Борисівна		
5	к. е. н., професор Небава Микола Іванович		

7 Дата видачі завдання 18.09.24

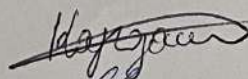
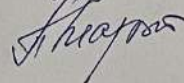
8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2— Календарний план

№ з/п	Назва етапів проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Пошук та огляд інформаційних джерел	03.10.2024р.	виконано
2	Огляд та аналіз існуючих підходів щодо пошуку і розпізнавання особи у відеозображенні	14.10.2024р.	виконано
3	Результати проведених досліджень та методів виявлення і розпізнавання особи у відеозображенні	25.10.2024р.	виконано
4	Проектування засобів виявлення і розпізнавання особи у відеозображенні	19.11.2024р.	виконано
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання особи у відеозображенні	02.12.2024р.	виконано
6	Попередній захист	18.11.2024р.	виконано
7	Перевірка «антиплагіат»	04.12.2024р.	виконано
8	Аналіз виконання роботи, висновки, додатки	05.12.2024р.	виконано
9	Оформлення пояснювальної записки та ілюстративного матеріалу	10.12.2024р.	виконано
10	Перевірка якості виконання магістерської роботи та усунення недоліків	12.12.2024р.	виконано

Студент

Керівник

Віталій КАРДАШ

Тетяна МАРТИНЮК

АНОТАЦІЯ

УДК 004.93

Кардаш В. О. Програмний комплекс для пошуку і розпізнавання особи у відеозображенні. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма Комп'ютерна інженерія. Вінниця: ВНТУ, 2024. — 118 с.

На укр. мові. Бібліогр.: 37 назв; рис. 16; табл. 8.

Магістерську кваліфікаційну роботу виконано розробці технології виявлення і розпізнавання особи у відеозображенні. Здійснено огляд існуючих методів та засобів виявлення та розпізнавання особи у відеозображенні, визначено їх переваги та недоліки. Розроблено технологію виявлення і розпізнавання особи у відеозображенні, в основі якої є послідовне поєднання детектора виявлення зони із обличчям людини і нейронної мережі для розпізнавання особи у обраній сцені цифрового зображення. Розроблено програмний комплекс виявлення і розпізнавання особи у відеозображенні на основі використання згорткової нейронної мережі. Виконано навчання нейронної мережі для виконання розпізнавання заданої кількості осіб. Проведено тестування створеного програмного застосунку для визначення його продуктивності. Проведено економічний розрахунок доцільності виконання нової розробки, обчислені фінансові затрати та визначено економічні переваги від впровадження запропонованого рішення у вигляді завершеного програмного продукту.

Ключові слова: відеопотік, детектор виявлення об'єкта, згорткова нейронна мережа.

ANNOTATION

Kardash V. O. Software system for searching and recognizing a face in a video image. Master's thesis in specialty 123 - computer engineering, computer engineering education program. Vinnytsia: VNTU, 2024. — 118 p.

In Ukrainian language. Bibliographer: 37 titles; fig.: 16; table 8.

The master's qualification work is devoted to the development of a technology for detecting and recognizing a person in a video image. A review of existing methods and means of detecting and recognizing a person in a video image was carried out, their advantages and disadvantages were determined. A technology for detecting and recognizing a person in a video image was developed, which is based on a sequential combination of a detector for detecting a zone with a human face and a neural network for recognizing a person in a selected scene of a digital image. A software complex for detecting and recognizing a person in a video image was developed based on the use of a convolutional neural network. The neural network was trained to recognize a given number of people. The created software application was tested to determine its performance. An economic calculation of the feasibility of implementing a new development was carried out, financial costs were calculated and economic benefits from implementing the proposed solution in the form of a completed software product were determined.

Key words: video stream, object detection detector, convolutional neural network.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ПОШУКУ І РОЗПІЗНАВАННЯ ОСОБИ У ВІДЕОПОТОЦІ.....	11
1.1 Штучний інтелект як засіб пошуку і розпізнавання особи	11
1.2 Системи відеоспостереження для пошуку та розпізнавання особи	18
1.3 Методи розпізнавання особи за зображенням обличчя	23
1.4 Нейронна мережа глибокого навчання	32
2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ОСОБИ	39
2.1 Формування послідовності пошуку та розпізнавання особи	39
2.2 Нейронні мережі для розпізнавання особи	44
2.3 Навчання нейронної мережі.....	52
3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ПОШУКУ ТА РОЗПІЗНАВАННЯ ОСОБИ	59
3.1 Формування архітектури програми розпізнавання особи.....	59
3.2 Розробка програми розпізнавання особи	62
3.3 Налаштування нейронної мережі	66
4 ТЕСТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ПОШУКУ ТА РОЗПІЗНАВАННЯ ОСОБИ	72
4.1 Навчання і перевірка роботи нейронної мережі	72
4.2 Аналіз якості роботи навченої моделі нейронної мережі	78
5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ РОЗПІЗНАВАННЯ ОСОБИ У ВІДЕОЗОБРАЖЕННІ.....	82
5.1 Комерційний та технологічний аудит науково-технічної розробки....	82

					08-54.МКР.007.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Кардаш В. О.			Програмний комплекс для пошуку і розпізнавання особи у відеозображенні. Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник		Мартинюк Т. Б.					6	118
Опонент		Черноволик Г.О.				ВНТУ, ар. 1КІ-23м		
Н. Контроль		Швець С.І.						
Затверджую		Азаров О.Д.						

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи	85
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	90
ВИСНОВКИ	97
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	99
ДОДАТОК А Технічне завдання	103
ДОДАТОК Б Методи розпізнавання обличчя людини	107
ДОДАТОК В Типи нейронних мереж	108
ДОДАТОК Г Структура програми розпізнавання особи	109
ДОДАТОК Д Лістинг програми виділення кадрів	110
ДОДАТОК Е Лістинг програми налаштування нейронної мережі	112
ДОДАТОК Ж Інтелектуальні методи розпізнавання	117
ДОДАТОК И Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	118

					08-54.МКР.007.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Фундаментальним завданням комп'ютерного зору є виявлення та розпізнавання об'єктів, яке має широке застосування у банківській сфері, криміналістиці, соціальних мережах, системах автономного водіння та інших системах спостереження [1-3]. Мета засобів комп'ютерного зору – це знайти та розпізнати об'єкти у отриманому зображенні або відео. Завдання це має дуже широкий спектр застосувань, таких як медична візуалізація для діагностики захворювань за допомогою рентгенівських знімків, виявлення пухлин у організмі людини, декодування ланцюгів ДНК та пошук нуклеотидів, в автоматизованих системах спостереження для розпізнавання обличчя людини, номерних знаків транспортних засобів або дорожніх знаків, у робототехніці для навігації роботів, маніпуляції об'єктами та виконання візуального контролю.

На теперішній час відома значна кількість методів розпізнавання особи за зображенням обличчя, таких як геометричний метод [4], метод головних компонент [5], використання фільтрів Габора [6] або прихованої Марківської моделі [7], метод порівняння шаблонів [8], метод Віюлі-Джонса [9], метод власних обличч [10], нейромережеві методи [11] та інші. Усі ці методи дозволяють із певною швидкістю та точністю здійснити пошук та розпізнавання особи. Кожен із цих методів має свої характеристики, що призводить до небажаних випадків «пропустити чужого або ж відмовити у доступі для свого».

Відомі методи виявлення об'єктів у комп'ютерному зорі мають різні переваги й недоліки, що значно ускладнює вибір оптимального підходу для конкретного застосування. Методи можуть мати обмежену точність у виявленні об'єктів або низьку швидкість обробки даних. Також існуючі методи можуть бути витратними із точки зору використання обчислювальних ресурсів або бути складними в реалізації, тому завжди слід виконати порівняльний аналіз різних підходів з метою визначення кращого методу для

конкретного використання в засобах комп'ютерного зору. Однак автоматичне розпізнавання особи за зображенням обличчя все ще стоїть перед багатьма проблемами.

Актуальність дослідження пов'язана з необхідністю покращення взаємодії людини із оточуючим середовищем для вирішення ряду задач відеоспостереження. Розпізнавання особи є частиною задачі розпізнавання об'єктів та є однією із головних задач у області використання комп'ютерного зору.

Об'єктом дослідження є процес обробки даних про об'єкти зображень шляхом пошуку і розпізнавання особи у відеозображенні.

Предметом дослідження є методи обробки цифрових зображень для пошуку і розпізнавання особи у відеозображенні.

Метою роботи є вдосконалення методів виявлення і розпізнавання обличчя людини у цифрових зображеннях.

Для досягнення цієї мети потрібно вирішити такі **задачі**:

- проаналізувати методи і засоби виявлення і розпізнавання обличчя людини у відеопотоці;
- запропонувати удосконалений метод оброблення цифрових зображень для виявлення і розпізнавання обличчя людини;
- розробити послідовність оброблення цифрового зображення для виявлення і розпізнавання обличчя людини;
- обрати інструментарій та написати програму для виявлення і розпізнавання обличчя людини;
- виконати обґрунтування доцільності впровадження нового наукового рішення, зробити розрахунок економічних затрат для реалізації програмних засобів пошуку та розпізнаванню об'єктів та проаналізувати економічні вигоди від впровадження нового програмного рішення.

Методи дослідження магістерської роботи: використовувалися методи теорії обчислювальних систем, системний аналіз, теорії масового

обслуговування, методи просторового диференціювання функцій, методи множення матриць, методи математичної статистики для обробки отриманих результатів розпізнавання. Також у роботі використано принципи об'єктно-орієнтованого програмування для реалізації запропонованого підходу.

Наукова новизна роботи полягає в наступному вдосконалено метод опрацювання цифрового зображення для виявлення і розпізнавання обличчя людини, який відрізняється від існуючих послідовним поєднанням детектора виявлення зони із обличчям людини і нейронної мережі для розпізнавання особи у обраній сцені цифрового зображення, що дозволило зменшити загальний час на опрацювання всього кадру відео зображення.

Практичне значення одержаних результатів:

- сформовано послідовність опрацювання цифрових зображень для виявлення і розпізнавання обличчя людини;
- створено програму оброблення цифрового зображення для виявлення і розпізнавання обличчя людини.

Апробація результатів роботи: здійснено доповідь на LIV Всеукраїнській науково-технічній конференції підрозділів ВНТУ.

За результатами виконання магістерської роботи **опубліковано** 1 наукову працю [12] <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/author/submission/22893>.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ПОШУКУ І РОЗПІЗНАВАННЯ ОСОБИ У ВІДЕОПОТОЦІ

Завдання пошуку і виявлення обличчя людини у отриманому зображенні часто є першим кроком у вирішенні задачі вищого рівня по розпізнаванню особи, виділенню деталей особи або ж її міміки. У даному розділі роботи розглянуті методи та засоби пошуку і розпізнавання особи у отриманому потоці відео даних.

1.1 Штучний інтелект як засіб пошуку і розпізнавання особи

В останні роки різко зріс інтерес до використання штучного інтелекту (ШІ, з англійської AI), до технологій глибинного машинного навчання та їх застосування у системах електронної безпеки. Ці технології на теперішній час активно використовуються в багатьох функціях відеоаналітики. Штучний інтелект допомагає розпізнавати особу за зображенням обличчя та автомобільні номери, допомагає здійснювати пошук і вести стеження із камер відеоспостереження за переміщенням окремо взятої людини та виконувати багато ще чого.

Сучасний штучний інтелект можна розподілити на три умовні типи систем: системи, що здатні самостійно приймати рішення; системи, що надають можливість забезпечення підтримки прийняття рішень; і системи, що здатні миттєво обробляти значні обсяги даних і вишукувати у них закономірності й відхилення від норми у вигляді аномалій.

Ось цей третій тип штучного інтелекту часто використовується в електронних системах безпеки. Цей тип штучного інтелекту може швидко обробляти, класифікувати і відтворювати великі обсяги відеоданих з камер спостереження. Багато можливостей інтелектуального аналізу відео в системах спостереження засновані на навичках цього ШІ. Беруться до уваги такі технології, як машинне навчання, комп'ютерний зір та глибокий аналіз даних.

Машинне навчання (наприклад, функції розпізнавання облич в IP-камерах безпеки - чим більше облич в базі даних пристрою, тим більше людей програма може ідентифікувати і знайти, коли отримує команду пошуку).

Комп'ютерний зір — виявлення, класифікація та відстеження певних типів об'єктів — наприклад, камера безпеки на автомагістралі може бути дистанційно запрограмована на виявлення червоного автомобіля, якщо за кермом перебуває злочинець [3]. Комп'ютерний зір — це сукупність науково-технічних елементів, що використовуються для створення комп'ютерних систем виявлення, класифікації та відстеження об'єктів. Комп'ютерний зір - це теорія і технологія створення штучних систем, які отримують дані у вигляді зображень, що можуть бути представлені у вигляді відеопослідовностей, зображень з різних камер спостереження або веб-камер. Одним з основних застосувань комп'ютерного зору є машинний зір.

Глибокий аналіз даних — це структурування, аналіз і вилучення корисної інформації з різномірних інформаційних потоків.

І одна із найцікавіших технологій штучного інтелекту, вже активно використовується у відеоспостереженні, це використання так званої «нечіткої логіки» або «розмиті логіки». У цій техніці не дають машині чіткого алгоритму для пошуку чогось, а лише загальний огляд контексту в надії, що машина навчиться розуміти поведінку людини в загальних рисах, що не завжди доречно і не завжди логічно [13].

Система відеоспостереження зі штучним інтелектом має ряд переваг, до яких відносяться підстрахування або ж і часткова заміна людей зі служби охорони, багатозадачність, робота із архівом та стратегічне планування.

Резервне або часткове дублювання для охоронців — за часів аналогового відеоспостереження зображення з камер передавалося на монітори, за якими безперервно спостерігала жива людина. Її завданням було вчасно виявити будь-яку неадекватну або злочинну поведінку в кадрі і швидко подати сигнал колегам для усунення потенційної небезпеки. Сьогодні всі види тривоги автоматично запускаються самим штучним інтелектом, без втручання

людини. Крім того, AI виконує такі складні функції, як виявлення забутих або загублених речей, наприклад, сумок, або виявлення переміщення (трекінг) розшукуваних осіб у зоні дії камер. Крім того, на відміну від живої людини, AI ніколи не втрачає уваги і не пропускає завдання.

Багатозадачність — тобто, окрім прямої функції відеоспостереження, відеокамери виконують низку інших завдань, пов'язаних з аналізом відео. Від збору статистики (наприклад, скільки людей проходить через КПП за день, скільки автомобілів зупиняється на заправці тощо) до дозволу або заборони в'їзду на територію, що охороняється, певним людям або транспортним засобам. Деякі моделі AI-відеокамер можуть навіть дистанційно вимірювати температуру тіла людини.

Найпоширеніша і найпопулярніша функція AI у відеоспостереженні — управління архівами. AI спеціалізується на роботі з великими базами даних і може чудово класифікувати, організовувати та називати окремі фрагменти. Іншими словами, раніше, щоб знайти певний фрагмент записаного відео, потрібно було сідати і перемотувати відеокасету [15]. Це забирало багато часу. Тепер же, якщо шукати за певними параметрами, AI миттєво знайде те, що потрібно. Наприклад, якщо ввести в пошук фотографію людини, AI перевірить весь відеоархів і знайде всі відеофрагменти, в яких з'являється ця людина.

Стратегічне планування — новітня функція AI, яка використовується у відеоспостереженні. Візьмемо, наприклад, магазин або цілий торговий центр. Якщо відеокамери зі штучним інтелектом розміщені правильно, вони можуть збирати статистику, наприклад, які зони в магазині є найбільш і найменш людними в певний час доби, які полиці заповнені товарами, до яких полиць не підходять покупці тощо. Крім того, теплові карти також показують шляхи потенційних покупців через магазин, вік і стать більшості тих, хто заходить в магазин, і середній час, який відвідувачі проводять в магазині.

Машинний зір, розділ комп'ютерного зору, використовується для моніторингу виробничих процесів за допомогою камер відеоспостереження та цифрових пристроїв вводу/виводу. Системи машинного зору використовують

камери відеоспостереження та відповідне програмне забезпечення для обробки зображень. Системи машинного зору часто призначені для виконання вузькоспеціалізованих завдань, таких як виявлення і підрахунок об'єктів, виявлення переміщення об'єктів у просторі, зчитування серійних номерів і виявлення дефектів у деталях. Основними перевагами систем машинного зору є те, що вони можуть працювати безперервно і на високій швидкості та мають необхідну точність вимірювань.

Машинний зір — це процес використання комп'ютерного зору в системах відеоспостереження та промисловому виробництві, що складається з загальноновживаних апаратних і програмних компонентів. Типова система машинного зору складається з таких компонентів:

- одна або кілька цифрових або аналогових відеокамер (кольорових або чорно-білих) і відповідне оптичне обладнання для отримання зображень;
- програмне забезпечення для обробки зображень (у випадку аналогових камер це включає завдання перетворення зображень у цифровий формат);
- процесор (сучасний персональний комп'ютер з багатоядерним процесором або внутрішнім процесором);
- спеціалізоване програмне забезпечення машинного зору, яке надає інструменти для створення окремих програмних додатків;
- датчики (зазвичай оптичні або магнітні), які синхронізують виявлення, збір та обробку зображень.

Таким чином, машинний зір стає частиною комп'ютерного зору в комп'ютерній системі. Комп'ютерна система — це комплекс інформації та технологій, призначений для обробки, зберігання, введення та виведення інформації. До складу комп'ютерних систем входять комп'ютери, пристрої введення/виведення інформації та сервери з відповідним програмним забезпеченням. Комп'ютерні системи можуть бути об'єднані в комп'ютерні мережі. Обмін інформацією здійснюється через локальні або глобальні

системи передачі даних через комп'ютерні мережі. Для опису таких систем використовуються технічні, документальні, організаційні, алгоритмічні, функціональні, програмні та інформаційні структури. Комп'ютерні системи можуть бути інтегровані з іншими інженерними технологіями для розширення своїх можливостей, а різноманітність і значна консолідація комп'ютерного обладнання може створити єдине середовище для управління різними процесами.

Типова комп'ютерна система складається із таких компонентів, як апаратура комп'ютера, операційна система і прикладне програмне забезпечення.

Комп'ютерне обладнання — основними частинами є центральний процесор (CPU, Central Processor Unit), який виконує команди комп'ютера; пам'ять комп'ютера, яка зберігає програми та дані; пристрої, які називаються пристроями введення/виведення або зовнішніми пристроями (пристрої введення/виведення), які вводять дані в комп'ютер і виводять результати роботи комп'ютерної програми у формі, зрозумілій користувачеві (людині або іншій програмі); і пам'ять комп'ютера, яка зберігає дані та дані. (Основними частинами є центральний процесор (CPU, Central Processor Unit), який виконує команди комп'ютера, пам'ять комп'ютера, яка зберігає програми та дані, пристрої введення/виведення або так звані зовнішні пристрої (пристрої вводу/виводу), які вводять дані в комп'ютер і виводять результати роботи комп'ютерної програми у формі, зрозумілій користувачеві (людині або іншій програмі)).

Операційна система — системне програмне забезпечення, яке є основним інструментом управління і керує процесом використання комп'ютерного обладнання користувачем і різними програмами;

Прикладне програмне забезпечення — програмне забезпечення, призначене для вирішення різних типів завдань. Прикладне програмне забезпечення включає в себе компілятори, графічні бібліотеки, системи

управління базами даних, ігрові програми та різні офісні програми. Порівняно з операційними системами, прикладне програмне забезпечення утворює наступний рівень і дозволяє вирішувати на комп'ютері різні прикладні завдання.

Одним із таких прикладних програмних продуктів є програма пошуку та розпізнавання облич у відеопослідовності. Створення такого продукту є одним із завдань створення системи розпізнавання образів.

За методами й засобами пошуку об'єктів системи розпізнавання образів поділяють на детерміновані, логічні, ймовірнісні, структурні, експертні та нейроноподібні системи [2, 14].

Детерміновані системи, які використовують «геометричні» міри близькості для побудови алгоритмів розпізнавання образів. Такі системи вимірюють відстань між кількісними властивостями об'єктів і стандартами класу.

Логічні системи Логічні системи, засновані на дискретному аналізі та обчисленні логічних виразів. Такі системи покладаються на використання логічних властивостей об'єктів для розв'язання булевих рівнянь, щоб знайти клас, до якого слід віднести такі об'єкти.

Імовірнісні системи: використовують імовірнісні методи в рамках теорії статистики для створення алгоритмів розпізнавання образів. Для того, щоб такі системи розпізнавання працювали, повинна існувати ймовірнісна залежність між обраними властивостями об'єкта та його класом.

Структурні або лінгвістичні системи розпізнавання призначені для побудови таких алгоритмів розпізнавання за допомогою речень. Кожне з цих речень описує структуру об'єкта через певні недистрибутивні елементи. Такі речення утворюють своєрідну спеціалізовану мову. У таких системах класифікація об'єктів виконується шляхом порівняння речень невідомого об'єкта з еталонними реченнями заздалегідь встановленого класу.

Експертні системи розпізнавання образів. Такі системи використовуються, коли існує інтуїтивний або евристичний метод або

інструмент для вирішення завдання, але немає точного алгоритму або розрахунку. Експертні системи складаються з набору комп'ютерних програм, які містять накопичені знання експертів у певній галузі, можуть класифікувати об'єкти в цій галузі і, за необхідності, можуть генерувати відповіді, поради та рекомендації, запитуючи додаткову інформацію про об'єкт, що аналізується.

Нейроподібні системи, що використовують нейронні моделі для розпізнавання образів.

Системи розпізнавання образів можна розділити на некеровані, керовані та самонавчальні системи залежно від їхнього відношення до навчання [4, 13].

Некеровані системи мають достатньо інформації для побудови апіорного словника та алфавіту ознак класів, а також для побудови визначень усіх класів мовою цих ознак. Тому, як перший крок, достатньо визначити відстані або кінцеві межі між цими класами ознак і сформулювати точні правила, які дозволяють розбити отриманий набір ознак на відповідні класи. Для побудови такої системи розпізнавання потрібні повні початкові апіорні знання про об'єкт.

У системах навчання розпізнавання на основі зображень початкової інформації достатньо лише для побудови апіорного алфавіту класів та апіорного словника ознак, але не для визначення класів мовою ознак. Наявна початкова інформація дозволяє віднести об'єкт до різних класів. «Викладач неодноразово подає системі розпізнавання навчальні зображення відомих класів, які вже були згенеровані. Якщо система розпізнавання дає неправильну відповідь, система змінює всі коефіцієнти підсилення вибраних сигналів за відомим алгоритмом, поки не отримає правильну відповідь. Після такої перевірки та навчання система розпізнавання починає працювати самостійно.

У системі, що самонавчається, початкових апіорних знань достатньо лише для побудови словника ознак, але не для будь-якої іншої класифікації заданого об'єкта. На ранніх стадіях побудови системи системі надається навчальна множина об'єктів, ідентифікованих словником ознак. Через недостатнє початкове знання про те, до якого класу належать об'єкти в цьому

початковому наборі, система не отримує жодних інструкцій. На етапі самонавчання інструкції замінюються набором правил, і система розпізнавання образів починає самостійно будувати класифікацію для подальшого використання. Метою такого самонавчання є накопичення достатньої кількості інформації для адекватного функціонування системи розпізнавання. Для досягнення такого результату необхідний зворотний зв'язок для корекції та покращення оброблених апріорних знань.

1.2 Системи відеоспостереження для пошуку та розпізнавання особи

Сьогодні банки, офіси, великі компанії та навчальні заклади немислимі без сучасних комп'ютерних систем контролю, захисту та управління доступом (СКУД). Такі системи покликані надійно захищати складні ресурси, наявні в компаніях та інших організаціях, від різних можливих загроз [16]. Однією з найбільш перспективних систем вважаються безконтактні СКУД. В основі роботи таких систем лежить ідея розпізнавання співробітників або користувачів за певним набором характеристик або персональною карткою, яка використовується для ідентифікації та аутентифікації співробітників [17].

Система комп'ютерного контролю — це сукупність апаратних, програмних, електричних, механічних та інших засобів, які дозволяють і гарантують доступ певних осіб на певні ділянки (будівлі, приміщення, території) або до певних технічних засобів, предметів чи обладнання (сейфи, персональні комп'ютери, транспортні засоби тощо), обмежуючи або забороняючи доступ особам, які не мають цих прав. Впровадження засобів контролю не тільки значно підвищує рівень безпеки в компанії чи організації, але й дозволяє швидко реагувати на поведінку відвідувачів та персоналу, вести облік робочого часу та керувати робочими графіками.

Кожен співробітник або постійний відвідувач організації чи компанії отримує картку або брелок з відповідним ідентифікатором (спеціальним електронним ключем), тобто індивідуальним кодом. Такі електронні ключі видаються шляхом реєстрації обраних осіб за допомогою самої системи.

Персональна електронна картка містить паспортні дані, фотографію (або відеозображення) та іншу інформацію про власника електронного ключа. Видана персональна електронна картка та код електронного ключа поєднуються між собою та зберігаються у спеціально організованій комп'ютерній базі даних.

За такою системою працює більшість ЦСК. Перед входом в будівлю або на об'єкт, що охороняється, встановлюється спеціальний зчитувач, який зчитує з картки код та інформацію про права доступу власника картки і передає ці дані контролеру системи. Кожному коду в системі відповідає інформація про права власника картки. Порівнюючи цю інформацію та аналізуючи умови, за яких користувач пред'являє картку, система приймає відповідне рішення. Контролер приймає такі рішення, як блокування або відкриття дверей (турнікетів, замків), активація сигналізації та переведення приміщення або будівлі в режим охорони. Всі факти, пов'язані з пред'явленням картки та пов'язаними з ними діями (тривогами, проходами тощо), фіксуються контролером і зберігаються в пам'яті комп'ютера. Інформація про події, викликані пред'явленням картки користувачем, може бути використана в подальшому для отримання відповідних звітів про робочий час співробітників, порушення трудової дисципліни тощо.

Існуючі системи контролю доступу можуть бути автономними або мережевими, в залежності від специфіки бізнесу. Автономні системи контролю доступу використовуються для розмежування доступу до окремих приміщень організації і не потребують протоколів. Вони прості та ідеально підходять для невеликих офісів з невеликою кількістю дверей. Недоліком таких систем є те, що вони не аналізують, хто і коли проходить через кожні двері в офісі.

Мережеві системи контролю доступу, на відміну від автономних, базуються на комп'ютерній мережі і можуть повністю контролювати і аналізувати переміщення персоналу всередині компанії (організації), аналізувати робочий час, регулювати бажаний режим доступу в приміщення,

ідентифікувати співробітників по фотографії або прямому розпізнаванню обличчя.

Комп'ютерні системи контролю доступу складаються з таких основних компонентів, як зчитувальні пристрої, контролери обробки та виконавчі пристрої. Пристрої зчитування можуть бути контактними (за допомогою сенсорних чіпів пам'яті), кодовими (за допомогою набору кодів доступу), безконтактними (за допомогою проксіміті-карт) або біометричними (за допомогою райдужної оболонки ока або відбитків пальців). Контролери обробки даних зберігають і обробляють дані про відвідувачів і використовуються для зв'язку з центральним комп'ютером. До виконавчих пристроїв належать ворота, турнікети, шлюзи та шлагбауми з електромагнітними або електромеханічними замками. Мережеві системи контролю доступу управляються спеціальним програмним забезпеченням і, в залежності від завдання і конфігурації системи, виконують такі функції, як видача індивідуальних карт доступу, облік робочого часу, обмеження доступу в приміщення, ідентифікація співробітників і виведення їх фотографій на пульт управління.

При виборі структури та обладнання системи особливу увагу слід приділити аналізу її основних характеристик. Такими характеристиками систем контролю доступу є: надійність роботи, вартість, швидкодія, час реєстрації користувачів, обсяг пам'яті системи, ймовірність помилки при відмові в доступі авторизованим користувачам (помилки 1-го типу), ймовірність помилки при наданні доступу неавторизованим користувачам (помилки 2-го типу), зловмисні дії: стійкість до зловмисних дій тощо.

Відомі сучасні СКУД за типом ідентифікатора (зображення обличчя, райдужної оболонки ока, відбиток пальця, безконтактна картка, код доступу, множинна ідентифікація), типом пристрою (на основі пристроїв контролю та зчитування, на основі терміналів доступу, на основі автономних електронних замків), умовами використання (для зовнішнього використання, для використання в приміщенні) та типом програмування (мережеві — об'єднана

мережа пристроїв доступу під управлінням єдиного програмного забезпечення; автономні — управління пристроями на вході).

Сучасні системи контролю доступу побудовані на основі біометричних технологій контролю та управління доступом і за принципом роботи схожі на рішення для входу в мережу. Біометричні характеристики людини у вигляді цифрового шаблону зберігаються в пам'яті турнікета або дверного замка, і потім при кожному проході людина повинна виконати процедуру біометричної ідентифікації (проголошення кодової фрази, сканування сітківки ока або пальця), щоб відкрити двері [17].

Термін «біометричні технології» включає в себе автоматизовані або автоматизовані методи розпізнавання людей на основі їх біологічних характеристик або симптомів. Основними компонентами біометричного методу ідентифікації є сканер для вимірювання конкретної біометричної характеристики людини та алгоритм, який дозволяє порівняти її з раніше записаною характеристикою того ж типу (біометричним шаблоном). Такі системи мають два режими роботи: ідентифікація (одне і багато порівнянь) і верифікація (одне і тільки одне порівняння). Теоретична різноманітність можливих біометричних схем (геометрія руки, райдужна оболонка ока, геометрія обличчя, сітківка ока, відбитки пальців) досить велика, але на практиці доступні лише деякі з них. Використовуються три основні методи: зображення обличчя (2D або 3D фотографії), розпізнавання відбитків пальців і розпізнавання райдужної оболонки ока [17].

Біометричні СКУД застосовують такі методи розпізнавання, як розпізнавання райдужної оболонки ока, обличчя, відбитків пальців, голосу та геометрії руки. Такі системи іноді використовуються для створення підсистем обліку робочого часу співробітників. У таких випадках використовуються традиційні системи біометричного сканування або біометричні замки з мережевою архітектурою.

Інтегрована СКУД поєднує в собі можливості різних систем. Це означає, що співробітникам організації достатньо лише один раз зареєструватися у

адміністратора системи доступу, і їм будуть автоматично призначені всі повноваження, необхідні для входу в приміщення на роботу, роботи в наданій корпоративній мережі та використання її ресурсів.

Отже, найважливіше, що необхідно визначити, це вибір технології розпізнавання облич, яка буде використовуватися. При цьому рішення повинно ґрунтуватися на поєднанні двох основних критеріїв: надійності помилкового розпізнавання і надійності безпомилкового розпізнавання.

Алгоритми біометричної ідентифікації мають два якісні показники: false rejection rate (FRR) — надійність безпомилкового розпізнавання, тобто в цьому випадку система не розпізнає відому людину (гарантуючи, що «людина» не буде пропущена); і false acceptance rate (FAR) — надійність помилкового розпізнавання людини, тобто ймовірність того, що система переплутає двох людей і розпізнає «незнайомця» як «свою людину». На практиці при побудові системи контролю доступу зменшення FAR завжди призводить до зменшення точності обраного методу, що фактично призводить до збільшення FRR [4, 19]. Конкретна величина помилки ідентифікації залежить від обраного підходу, налаштування алгоритму ідентифікації та якості реалізації.

Зручність використання системи. Необхідно забезпечити, щоб співробітники організації могли комфортно та зручно проходити процедуру біометричної ідентифікації в рамках своєї роботи.

Стійкість обраного методу до впливу навколишнього середовища. Деякі робочі характеристики різних біометричних методів сильно залежать від навколишнього середовища, і їх стабільність може змінюватися при зміні цих умов. Наприклад, робота 2D-методів розпізнавання обличчя сильно залежить від навколишнього освітлення, а сканери відбитків пальців постійно забруднюються, що знижує їх продуктивність.

Стійкість систем до шахрайства. Біометричні системи повинні мати певний рівень стійкості до спуфінгу (несанкціонованого доступу). Щоб отримати несанкціонований доступ до системи, яка використовує відбитки пальців, може бути достатньо нанести графітовий порошок на зображення, що

цікавить, і пропустити його через тонку плівку. Комп'ютерні системи розпізнавання, засновані на двовимірних (2D) зображеннях обличчя людини, можна легко «обдурити», надавши системі фотографію знайомої людини [20].

Вартість системи. Оцінюючи фактори, пов'язані з реалізацією системи розпізнавання, слід враховувати, що вартість цього процесу складається з низки елементів. Наприклад, у випадку з мережевими системами безпеки - це сервер аутентифікації та ліцензія користувача, зчитувач, супутнє програмне забезпечення та послуги підтримки, а в деяких випадках - розробка окремих модулів інтеграції зі спеціалізованим корпоративним програмним забезпеченням.

Швидкість роботи біометричних систем. Цей критерій дуже зрозумілий. Чим швидше система розпізнає та ідентифікує користувача, тим краще. Слід зазначити, що швидкість обробки залежить від вибору методу розпізнавання - ідентифікації або верифікації. Це пов'язано з тим, що порівняння одного шаблону з усією базою даних реєстраційних шаблонів займає набагато більше часу, ніж порівняння один до одного.

Використання біометричних технологій поступово переходить від заміни інших систем ідентифікації співробітників (паролів, карток тощо) до простого поля конкуренції між різними методами біометричної ідентифікації. Наразі тривимірні зображення обличчя, відбитки пальців та райдужна оболонка ока є трьома найточнішими методами розпізнавання. З іншого боку, розпізнавання обличчя на основі двовимірних зображень обличчя та інші відомі біометричні методи (наприклад, розпізнавання голосу, розпізнавання на основі геометрії руки) поступаються цим методам розпізнавання з точки зору точності.

1.3 Методи розпізнавання особи за зображенням обличчя

На теперішній час відома значна кількість методів розпізнавання особи за зображенням обличчя, таких як геометричний метод, метод головних компонент, використання фільтрів Габора або прихованої Марківської моделі,

метод порівняння шаблонів, метод Віоли-Джонса, метод власних обличь, нейромережеві методи та інші. Усі ці методи дозволяють із певною швидкістю та точністю здійснити пошук та розпізнавання особи. Кожен із методів має свої переваги та недоліки та певну сферу застосування. Розглянемо відомі методи розпізнавання обличчя людини для статичних двовимірних зображень. Усі ці методи розпізнавання обличчя людини за способом подання зображення обличчя можуть бути розподілені на такі 3 класи, як методи зіставлення в цілому, методи зіставлення за ознаками та гібридні методи [14].

Найбільш широко використовуваним класом методів є група методів зіставлення в цілому. Для цієї групи методів вся область обличчя розглядається в якості вихідних даних для здійснення процесу розпізнавання. Компоненти та сегменти обличчя не аналізуються окремо. Інші особливості, такі як кут, під яким розглядається обличчя, або вираз обличчя, не беруться до уваги. Першим і на сьогодні найпоширенішим методом цієї групи є метод Eigenface [10]. Всі методи цієї групи використовують аналіз головних компонент [4], лінійний дискримінантний аналіз [5], аналіз незалежних компонент [7] та вейвлет-перетворення (особливо вейвлети Габора) [6] для створення проєкції вхідного зображення на підпростір (Додаток Б до магістерської роботи). Основною перевагою цього класу методів є те, що вони не відкидають інформацію із зображення обличчя і зосереджуються лише на обмежених областях або характерних точках. Однак, це також призводить до недоліків. Ці методи є обчислювально дорогими, вимагають високої кореляції між тестовим і навчальним зображеннями і не є ефективними, коли існують великі відмінності в позі, масштабі та освітленні.

Наступна група методів — це методи зіставлення ознак або структурні методи. Цей клас методів використовує вибрані риси обличчя та їх структуру як вхідні дані для розпізнавання. У цих методах з виявлених областей обличчя спочатку вибираються характерні об'єкти, такі як очі, ніс, рот, брови і підборіддя. Потім обчислюються структурні характеристики (геометричні

розміри, пропорції або особливості зовнішнього вигляду) обличчя. Таким чином, вихідне зображення представляється вектором структурних ознак. Існують різні типи ознак, що використовуються для розпізнавання, такі як ключові області, ключові точки та контури [4]. Основною проблемою цього класу методів є завдання відновлення рис обличчя, які неможливо знайти на зображенні, наприклад, коли частини обличчя приховані [14]. До цієї групи належать, наприклад, такі методи, як:

- методи, засновані на геометричних особливостях обличчя [4]: ідентифікується набір ключових точок (або областей) обличчя, а потім вибирається набір ознак, де кожна ознака є або відстанню між ключовими точками, або відношенням їхніх відстаней (емпірично підбирається найбільш інформативний набір ознак);

- еластична контурна модель обличчя [5]: розпізнавання базується на порівнянні контурів обличчя, таких як голова, вух, губ, носа, брів та ліній очей;

- порівняння пружних (деформованих) графів [6]: обличчя представляються у вигляді графів з вершинами в ключових точках, таких як контури голови, губ і носа або їх полюси.

Основною перевагою цього класу методів є те, що вони стійкі до змін положення обличчя на вхідному зображенні та інваріантні до розміру, орієнтації та освітлення. Ще однією перевагою є компактність представлення зображення обличчя та висока швидкість зіставлення. Основними недоліками таких підходів є складність автоматичного вибору найкращого набору ознак для розпізнавання та висока обчислювальна складність.

Ідея гібридних методів прийшла із аналогії із людським зором, де розглядаються як локальні характеристики обличчя людини, так і усе обличчя в цілому. Цей клас методів використовує комбінацію загальних методів і методів зіставлення на основі ознак. Також, залежно від методу розпізнавання, всі методи можна розділити на дві групи: статистичні та інтелектуальні методи.

Статистичні підходи розпізнають вхідні зображення шляхом обчислення статистичних порівнянь між зразками в базі даних і еталонними зображеннями. Друга група методів використовує нейронні мережі, приховані марковські моделі та методи машинного навчання для розпізнавання або вилучення облич (Додаток Ж до магістерської роботи).

Лінійний дискримінантний аналіз (Fisherfaces, дискримінантний метод Фішера) [6] використовує проекцію простору зображень на простір ознак для групування найближчих зображень з однаковим обличчям і одночасного видалення зображень з різними обличчями. Як і раніше, цей метод значно покращує кластеризацію (розділення) зображень і зменшує кількість ознак. Алгоритм припускає, що для кожної особи в базі даних є велика кількість фотографій людини за різних умов освітлення. Переваги: завдяки набору фотографій кожної людини алгоритм стійкий до змін умов освітлення. Недоліки: немає інваріантності до афінних перетворень. Метод дозволяє досягти рівня розпізнавання до 99% за дуже різних умов.

Алгоритми на основі штучних нейронних мереж. Перевага використання штучних нейронних мереж для розпізнавання облич полягає в тому, що систему можна навчити ідентифікувати основні риси обличчя за допомогою навчальних вибірок. Нейронні мережі дозволяють створити класифікатор, який успішно моделює складну функцію розподілу зображень обличчя. Завдання класифікації передбачає непряму ідентифікацію основних ознак в мережі, визначення важливості ознак і системи взаємозалежності між ними. Серед нейронних мереж для розв'язання задач розпізнавання образів найбільш широко використовуються багат шарові перцептрони, мережі з радіальними базисними функціями та згорткові нейронні мережі, що використовують зворотне поширення помилок [18]. Нейронні мережі можуть бути успішно застосовані для розпізнавання облич і зіставлення облич на зображеннях. Однак зі збільшенням кількості класів, які потрібно оцінити, якість розпізнавання знижується.

Даний підхід має ряд недоліків, таких як додавання нового еталонного обличчя в базу даних вимагає повторного повного перенавчання нейронної мережі на такому новому наборі; процедура навчання є досить тривалою й важко формалізується та складний етап вибору архітектури мережі, до якого входить вибір кількості нейронів, шарів та характеру зв'язків між ними.

Метод Віоли-Джонса. В даний час метод Віоли-Джонса [9] є одним із найпопулярніших методів, який використовується для пошуку областей обличчя на зображеннях завдяки своїй швидкості та ефективності. Цей метод дозволив реалізувати виявлення облич у реальному часі в таких практичних додатках, як цифрові камери та програмне забезпечення для управління фотоальбомами.

Детектор облич Віоли-Джонса базується на трьох основних принципах: інтегроване представлення зображення на основі ознак Хаара, що дозволяє дуже швидко обчислювати необхідні ознаки; використання методу побудови класифікатора на основі адаптивного алгоритму бустінгу (AdaBoost); та метод об'єднання класифікаторів у каскадну структуру. Ці ідеї дозволяють швидко знаходити обличчя на зображеннях у реальному часі.

Ознака Хаара — це функція цифрового зображення, яка використовується для розпізнавання зображень. Названа через інтуїтивну схожість з вейвлетом Хаара; вейвлет Хаара є одним з перших і найлегших вейвлетів. Він був запропонований угорським математиком Альфредом Хааром. Ці вейвлети є ортогональними, тобто компактними, добре локалізованими в просторі, але не гладкими. Функції Хаара в основному використовувалися в перших детекторах обличчя в реальному часі; Віола і Джонс реалізували ідею використання вейвлетів Хаара і розробили те, що вони назвали функціями Хаара.

Найпростіша прямокутна ознака Хаара може бути розпізнана як різниця між сумами пікселів двох сусідніх областей у прямокутнику. Цей тип ознак називається 2-прямокутною ознакою. Кожна функція може точно показати, де знаходяться межі між темними і світлими областями.

До переваг відноситься висока швидкість виявлення об'єктів та висока ймовірність точного виявлення обличчя людини (понад 90 %) для фронтальних зображень обличчя, а також спостережень обличчя під невеликим кутом зору (до 30°) і низька ймовірність помилкового виявлення обличчя.

До недоліків відноситься дуже великий час навчання і значна чутливість до умов освітлення, а також при великому куті нахилу голови людини ймовірність виявлення обличчя різко падає.

Зіставлення за шаблоном. В основі цієї методики [8] лежить виділення ділянок обличчя на зображенні та порівняння їх для двох різних зображень. Кожна співпадаюча область збільшує схожість між зображеннями. Для порівняння областей використовуються найпростіші алгоритми, такі як попиксельне порівняння. Недоліком цього методу є те, що він вимагає багато ресурсів як для зберігання, так і для порівняння регіонів.

Оскільки використовуються найпростіші алгоритми порівняння, зображення повинні бути зняті в строго визначених умовах: не допускаються помітні зміни ракурсу, освітлення, емоційного виразу обличчя тощо. Точність розпізнавання цього методу становить близько 80%, що є хорошим результатом.

Марковські моделі можна назвати дуже потужним інструментом для розпізнавання образів і моделювання різних процесів. Це статистична модель, яка імітує поведінку процесу, подібного до марковського процесу з невідомими параметрами. Елементами моделі є початкова ймовірність стану, набір прихованих станів, набір спостережуваних станів і матриця ймовірностей переходів. При розпізнаванні об'єкта перевіряється марковська модель, створена для даної бази даних об'єктів, і обчислюється максимальна ймовірність спостереження, де послідовність спостережень цього об'єкта генерується відповідною і необхідною моделлю. Недоліком марковських моделей є те, що ці моделі не володіють дискримінантною здатністю. Це означає, що алгоритм навчання може лише максимізувати реакцію кожної

моделі на свій клас, але не може водночас мінімізувати її реакцію на інші класи або виділити важливі ознаки, які відрізняють один клас від іншого.

Одним із перших і до теперішнього часу основних методів розпізнавання особи виділяють метод виявлення геометричних характеристик обличчя [4]. Суть методу ґрунтується на тому, що на обличчі людини позначаються ключові точки. Після цього етапу визначаються відстані, власне це і є геометричними характеристиками обличчя між потрібними ключовими точками. Такі точки на зображенні обличчя людини можуть бути розташованими у різних місцях. Для прикладу вони можуть бути розміщені на краю ока, на губі, на носі, на мочці вуха і т. д. Як представлено на рисунку 1.1 такі ідентифікаційні точки обличчя позначенні наступним чином — центри брів (1а, 1б), верхні крайні точки вух (3а, 3б), центри зіниць (2а, 2б), правий кут правого і лівого ока (4а, 4б), кінець мочок вух (6а, 6б), ліві кути очей (5а, 5б), крайні точки носа по горизонталі (7а, 7б), кути рота (9а, 9б), кінчик носа (8), центр рота (10), та точку перетину ліній, яка розділяє верхню й нижню губу обличчя людини, та перпендикуляра, що проведений із точок, що позначають кінчики носа та підборіддя (11).

Можуть також ще виділятися і інші відстані на обличчі. Відстані (1), (2), (5), (4), (6), (8), (7), (11) можна вважати як основні, так як дія на них зі сторони таких чинників, як зачіска, макіяж, прикраси та інші фактори є доволі незначними. Введення ознак обличчя у форматі відношення відстаней для створення ідентифікаційних одиниць робить їх незалежними від масштабу по відстані, із якої здійснюється фотографування обличчя людини. У тому випадку, коли застосовується реальна величина розмірів голови людини або розміри її ділянок практично неможливо визначити, ну а для формування таких ідентифікаційних ознак абсолютно не є важливим, на якій відстані від об'єктива розташовується людина під час зйомки.

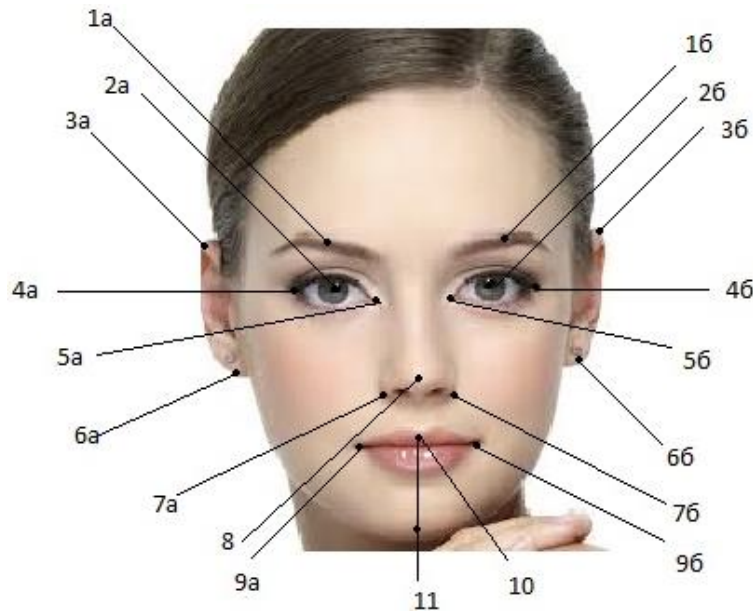


Рисунок 1.1 — Точки на фронтальній проекції обличчя людини

Методи аналізу головних компонент (РСА) використовуються для стиснення інформації без значної втрати інформації. Спочатку методи головних компонент застосовувалися в статистиці для зменшення простору ознак без значної втрати даних. У задачах розпізнавання обличчя він в основному використовується для представлення зображення обличчя у вигляді вектору низької розмірності або головної компоненти і порівняння його з еталонним вектором, що зберігається в базі даних. Основна мета методу головних компонент — значно зменшити розмірність простору ознак і краще ідентифікувати «типові» зображення, що належать до заданого набору облич. Метод використовується для виявлення різних варіацій у навчальному наборі зображень облич та ідентифікації цих варіацій на основі деяких ортогональних векторів, які називаються власними поверхнями. Набір власних векторів, отриманий один раз на навчальному наборі зображень облич, використовується для кодування всіх наступних зображень облич, представлених зваженою комбінацією цих власних векторів. Метод головних компонент полягає в наступному. Спочатку весь навчальний набір облич перетворюється на загальну матрицю даних, де кожен рядок представляє екземпляр зображення обличчя, який було згенеровано та розкладено. Всі

зображення в навчальному наборі повинні бути приведені до однакового розміру і мати нормалізовану гістограму.

Суть методу порівняння графів полягає в гнучкому порівнянні графів, що описують зображення обличчя. Зображення подаються у вигляді графа зі зваженими вершинами та ребрами. На етапі розпізнавання один граф, еталонний, залишається незмінним, тоді як інший граф деформується, щоб найкраще відповідати початковому графіку. У подібних системах розпізнавання граф може бути представлений як прямокутна сітка або як структура точок вимірювання обличчя людини [6]. Значення ознак обчислюються локально у вершинах графа, зазвичай за допомогою вейвлетів Габора (або структур Габора), в заданій локальній області вершин графа шляхом згортки комплексних значень або впорядкованого набору фільтра Габора, тобто значень інтенсивності пікселів, за допомогою фільтра Габора [6]. Ребра графа зважуються за відстанню між сусідніми вершинами. Ця різниця, яка є відмінною рисою між двома графами, обчислюється за допомогою спеціальної функції вартості деформації, яка враховує як різницю значень ознак, обчислених у вершинах, так і ступінь деформації ребер графа (рис. 1.2).

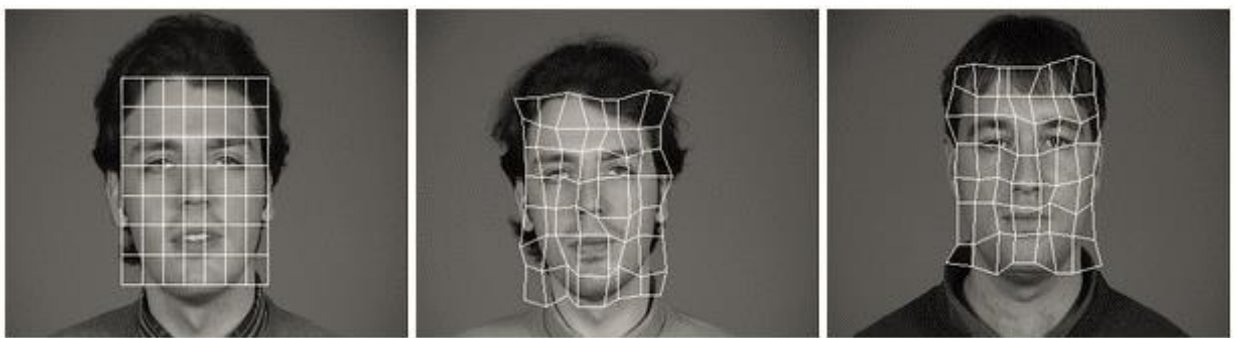


Рисунок 1.2 — Деформація графа у вигляді регулярної решітки

Граф деформується шляхом зсуву кожної вершини на певну відстань у потрібному напрямку від її початкового положення і вибору такого положення, при якому різниця між значеннями ознак у вершинах деформованого графа і значеннями ознак у відповідних вершинах еталонного

графа буде якомога меншою. Цей процес виконується для всіх вершин графа по черзі до тих пір, поки сума різниць значень ознак деформованого та еталонного графа не буде мінімальною.

Одні з найкращих результатів у розпізнаванні облич були отримані за допомогою згорткових нейронних мереж (CNN) [21] (рис. 1.3). Цей успіх пояснюється їхньою здатністю враховувати 2D топологію зображення, на відміну від того ж багатоваріантного перцептрона [14].

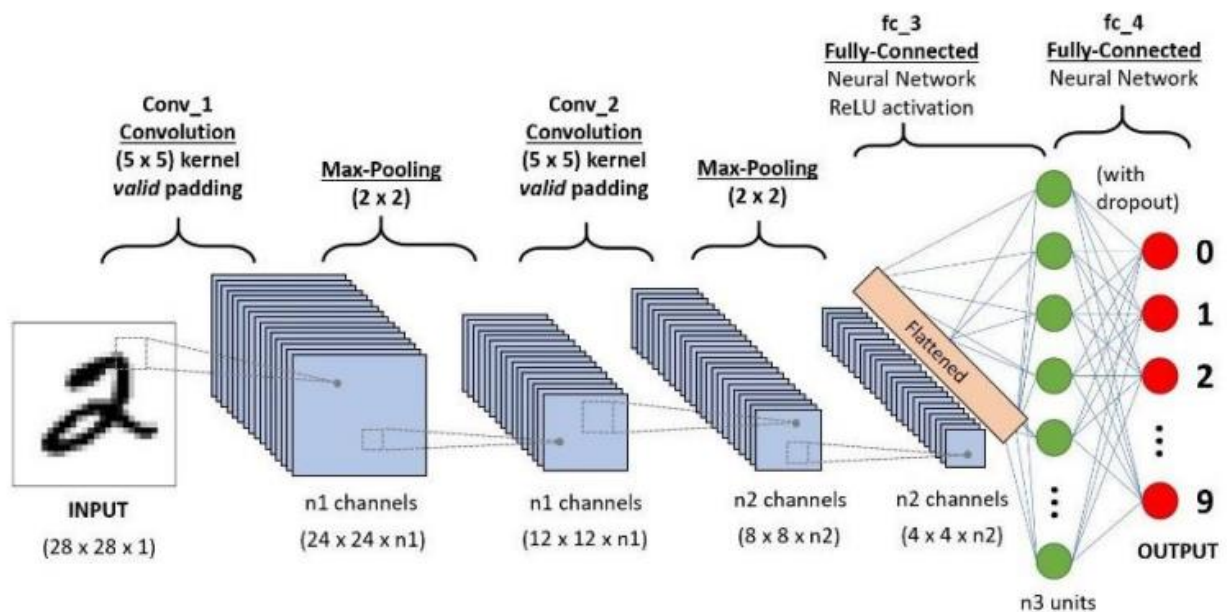


Рисунок 1.3 — Згорткова нейромережа

Такі нововведення забезпечують ШНМ часткову толерантність до зміни масштабу, поворотів, зсувів, кутових змін та інших спотворень. До недоліків нейромережевих методів можна віднести необхідність додавання нових референтних осіб до бази даних, що вимагає повного перенавчання мережі на всьому доступному наборі, що є досить тривалою процедурою і може зайняти значний час залежно від обсягу вибірки.

1.4 Нейронна мережа глибокого навчання

Операції пошуку, виділення та розпізнавання особи за зображенням обличчя у відео послідовності можна виконати за допомогою нейронної мережі глибокого навчання. На теперішній час відома значна кількість

нейронних мереж глибокого навчання, таких як згорткові нейронні мережі, глибокі короткі мережі (DBN), рекурентні нейронні мережі (RNN), глибокі нейронні мережі (DNN) та інші типи нейронних мереж. Кожна із має свої переваги та недоліки і сфери використання. Однією із найбільш поширених є згорткова нейронна мережа.

Згорткові нейронні мережі (ЗНМ) — це спеціалізована архітектура штучних нейронних мереж, призначена для ефективного розпізнавання зображень і є частиною інструментів і методів глибокого навчання [18]. Основою побудови згорткової нейронної мережі є чергування згорткових шарів і шарів субдискретизації (субдискретизації або об'єднання шарів і шарів субдискретизації). Структура згорткової нейронної мережі є односпрямованою (без зворотного зв'язку) і, по суті, багат шаровою. Для навчання мережі використовуються добре відомі стандартні методи, як правило, з використанням методів зворотного поширення.

Глибокі нейронні мережі мають багато прихованих шарів між вхідним і вихідним шарами, що називається «глибиною» нейронної мережі. Ці приховані шари дозволяють моделі автоматично ідентифікувати важливі особливості та закономірності у вхідних даних і вирішувати складні завдання, такі як розпізнавання об'єктів на зображенні, обробка природної мови, генерація тексту та багато інших завдань машинного навчання.

Процес згортання в згорткових нейронних мережах використовує лише кінцеву вагову матрицю невеликого розміру, яка переміщується через весь шар обробки (на самому початку процесу, безпосередньо над вхідним зображенням) і після кожного переміщення виробляє сигнал активації, що відповідає кожному нейрону в наступному шарі в тій же позиції. Тому для різних нейронів у вихідному шарі мережі використовується однакова вагова матриця, яку також називають ядром згортки. Ця матриця інтерпретується як графічне кодування ознаки, наприклад, лінія під певним кутом. Потім, у наступному шарі, який є результатом згорткового процесу, ця вагова матриця

вказує на наявність цієї ознаки та її координати в обробленому шарі і створює карту, яка називається картою ознак.

У згортковій нейронній мережі існує не просто набір ваг, а цілий набір ваг, що кодують елементи зображення (наприклад, дуги і лінії під різними кутами). Такі ядра згортки не задаються дослідником заздалегідь, а генеруються мережею самостійно в процесі навчання за допомогою відомого класичного методу зворотного поширення помилок. Кожен шлях з таким набором ваг створює власну копію карти ознак, що робить нейронну мережу багатоканальною (з багатьма незалежними картами ознак в одному шарі). Слід також зазначити, що при обробці шару з ваговою матрицею він переміщується не повними кроками (відповідно до розміру цієї матриці), а лише на невелику відстань (на один-два нейрони або пікселі).

Наступний процес субдискретизації (Subsampling або Pooling, що перекладається як «додискретизація» або «об'єднання») зменшує розмір згенерованої карти ознак. У цій архітектурі нейронної мережі дані про наявність ознаки вважаються більш важливими, ніж точне знання координат шуканого об'єкта, тому з декількох сусідніх нейронів на карті ознак вибирається максимальне значення ознаки і отримується стисла карта ознак меншого розміру, яка об'єднується в один нейрон. Цей процес не тільки прискорює подальші обчислення, але й робить нейронну мережу більш інваріантною до масштабу вхідного зображення.

Структура типової згорткової нейронної мережі складається з багатьох шарів. Після того, як вхідне зображення досягає першого шару, сигнал проходить через серію згорткових шарів, де по черзі виконується згортка і субдискретизація (об'єднання). Чергуючи ці шари, можна побудувати «карту ознак» з карт ознак кожного шару, причому кожен наступний шар зменшує розмір карти, але збільшує кількість оброблюваних каналів. На виході згорткового шару нейронної мережі створюється ще кілька шарів повністю пов'язаної нейронної мережі (у вигляді персептронів), які подаються на вхід

остаточної карти ознак. Типова архітектура згорткової нейронної мережі показана на рисунку 1.4 [18].

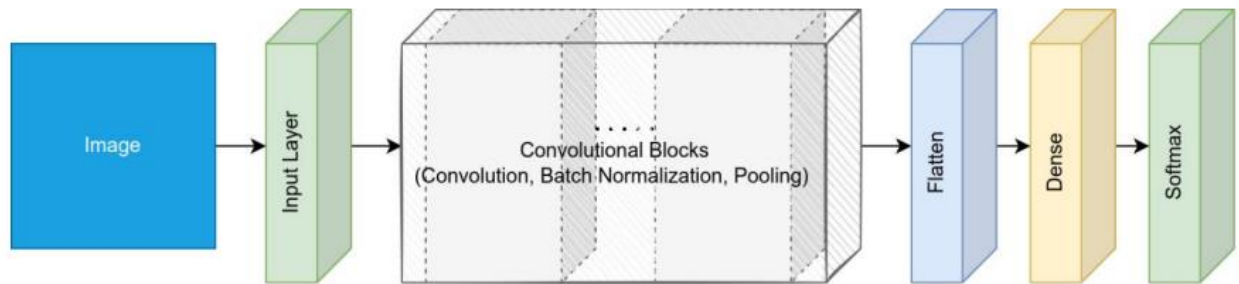


Рисунок 1.4 – Структура згорткової нейронної мережі

В згортковому шарі нейронної мережі над зображеннями, що подаються на вхід для обробки, здійснюється операція згортки [18]:

$$(f*g)[m,n] = \sum_{t,r} (f[m-t, n-r] \times g[t,r]), \quad (1.1)$$

де f — початкова матриця зображення, що підлягає обробці;

g — ядро згортки;

t — порядковий номер рядка ядра згортки;

r — порядковий номер стовпця ядра згортки.

Виконання такої операції згортки продемонстровано на рис.1.5.

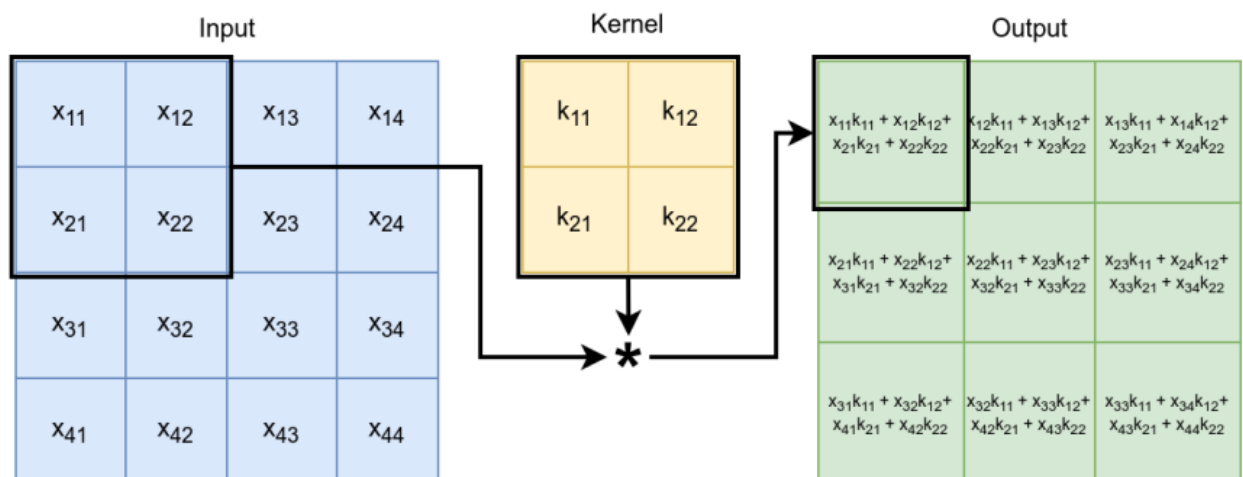


Рисунок 1.5 — Операція згортки

Ядро згортки — це набір вагових коефіцієнтів. Результатом цього процесу є зображення, яке називається картою ознак. Залежно від коефіцієнтів

обраного ядра згортки, карта ознак виділяє певні ознаки вихідного вхідного зображення.

Щоб виділити найбільш повні та якісні ознаки вихідного зображення, використовується декілька різних ядер згортки, в результаті чого на виході шару згортки отримується декілька карт ознак. Ядро згортки - це фільтр або вікно, яке проходить через всю попередню карту і знаходить специфічні особливості об'єкта зображення.

Скалярний результат процесу згортки надсилається до функції активації. Раніше для цього традиційно використовувалися гіперболічні тангенс і сигмоїдальні функції. Зараз більш поширеними є функції ReLU, які завдяки своїй простоті прискорюють процес навчання мережі та спрощують обчислення.

Наступний рівень піддискретизації — це згущення нелінійної карти ознак, де група пікселів (зазвичай 2×2) після нелінійного перетворення ущільнюється в один піксель. Найчастіше для цього використовується функція максимальної правдоподібності. У цьому перетворенні використовуються непересічні прямокутники або квадрати, кожен з яких перетворюється в один піксель і вибирається піксель з найбільшим значенням. Фільтрація непотрібних деталей зображення дозволяє уникнути перенавчання мережі. Як правило, шар субдискретизації розміщується після шару згортки і перед наступним шаром, який виконує згортку.

Після декількох операцій, які проходять через згортку зображення і подальше його стиснення за допомогою операцій піддискретизації, система змінює форму зображення з конкретної сітки початкових пікселів з високою роздільною здатністю для створення більш абстрактної карти особливостей. У кожному наступному шарі мережі кількість каналів збільшується, а розмірність аналізованого зображення для кожного каналу зменшується. В результаті залишається велика кількість каналів і зберігається невелика кількість даних або окремих параметрів, які інтерпретуються як найбільш абстрактні поняття, присутні на вихідному зображенні.

Ці дані об'єднуються і передаються в повністю пов'язану нейронну мережу. У цьому випадку повністю зв'язаний шар мережі вже втратив свою початкову просторову піксельну структуру і має менший розмір порівняно з кількістю пікселів у вихідному зображенні.

Мережі зазвичай реалізуються у вигляді комп'ютерних програм, але все більше чіпів реалізують нейронні мережі апаратно. Важливою особливістю мереж є їхня здатність до навчання. Глибоке навчання — це набір алгоритмів, які намагаються моделювати високорівневі абстракції в даних, використовуючи архітектуру багатьох нелінійних перетворень. У процесі навчання глибокої нейронної мережі отримана модель намагається представити об'єкти у вигляді комбінації простих примітивів (наприклад, у задачі розпізнавання обличчя такими примітивами є частини обличчя: ніс, очі, рот і т. д.). Додаючи шари, можна створювати більш високі рівні абстракції, що дозволяє моделям розпізнавати складні об'єкти реального світу.

Глибоке навчання вимагає великих обсягів навчальних даних і потужних обчислювальних ресурсів, таких як графічні процесори (GPU) і спеціальні тензорні процесори (TPU). Поширені архітектури глибоких нейронних мереж включають нейронні мережі прямого поширення, згорткові нейронні мережі для обробки зображень і рекурентні нейронні мережі для послідовних даних.

Глибокі нейронні мережі можна навчати за допомогою традиційних алгоритмів зворотного поширення. До цих алгоритмів було внесено багато вдосконалень. Таким чином, можна використовувати більше одного правила зважування.

Дві основні проблеми з глибокими нейронними мережами — це ті ж самі проблеми, які виникають при навчанні звичайних нейронних мереж: час навчання та час перенавчання.

Зворотне поширення та градієнтний спуск часто використовуються для навчання глибоких нейронних мереж через їхню простоту реалізації та хорошу збіжність. Однак при навчанні глибоких структур виникає кілька проблем, таких як кількість обчислювальних елементів, початкові умови для

ваг мережі та вищезгадані константи налаштування розміру кроку, які особливо важливі при оптимізації функцій у високорозмірних просторах.

Крім того, алгоритми стохастичного градієнтного спуску відомі для проблеми зникаючого градієнта, де градієнт, а отже, і швидкість навчання, слабшає, коли мережа заглиблюється від останнього до першого шару мережі. Тому навчання в більш глибоких шарах мережі відбувається дуже погано. Однак, останнім часом спостерігається тенденція до використання нелінійних функцій типу ReLU (Rectified Linear Unit) замість сигмоїдальних функцій активації вузлів, які можна записати як $\max(0, x)$ в глибоких мережах. Глибокі мережі з таким типом функції активації не страждають від градієнтних спотворень і можуть бути успішно навчені методом градієнтного спуску.

Виходячи із проведеного аналізу відаємо перевагу методу розпізнавання особи на основі нейронних мереж. Цей метод дозволяє після навчання сформованої мережі виконати розпізнавання особи на основі наявної бази даних обличч різних людей. Хоча процес навчання є досить тривалим, у підсумку отримуємо нейронну мережу, яка здатна із високою достовірністю виконувати завдання розпізнавання особи. Тип мережі, особливості її побудови, налаштування та навчання розглянемо у наступних розділах даної роботи.

У розділі був проведений аналіз сучасного стану сфери машинного зору по виділенню та розпізнаванню особи за зображення обличчя. Був виконаний аналіз таких методів, як геометричний метод, порівняння еластичних графів, метод Віюлі-Джонса, фільтри Габора, приховані Марковські моделі, метод порівняння шаблонів, метод головних компонент та використання нейронних мереж. Обрано методи на основі нейронних мереж, які зможуть більш вдало реалізувати завдання виділення та розпізнаванню особи за зображенням обличчя.

2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ОСОБИ

Робота комп'ютерних систем виділення і ідентифікації особи ґрунтується на виконанні пошуку та розпізнаванню обличчя людини у отриманому відео потоці. Процес виявлення обличчя людини у відео потоці здійснюється за послідовний ряд дій, що завершуються розпізнаванням особи із використанням сформованої бази даних осіб. Розробці послідовності та вибору засобів для виділення та розпізнавання обличчя людини у отриманій відеопослідовності присвячений даний розділ роботи.

2.1 Формування послідовності пошуку та розпізнавання особи

Серед біометричних методів ідентифікації особистості, які набули широкого розповсюдження і традиційно використовуються в багатьох галузях, найбільш поширеним і перспективним є метод розпізнавання обличчя. Цей метод розпізнавання має багато важливих і незаперечних переваг перед багатьма іншими методами, таких як можливість ідентифікації з досить високою точністю, необхідна верифікація на значній відстані, можливість прихованої верифікації, а також той факт, що потрібна лише відеокамера з необхідними характеристиками. На сьогоднішній день розроблено низку різних алгоритмів, які не тільки гарантують високу продуктивність і необхідну точність виявлення, але й дозволяють системі ідентифікації працювати в найрізноманітніших і складних умовах.

Для отримання бажаної точності розпізнавання особи можна виконати об'єднання декількох різних алгоритмів із набору біометрії, які виконують аналіз та формують набір ознак для розпізнавання особи за зображенням обличчя. Однак слід відмітити, що не у всіх випадках є доцільним використання значної кількості різних алгоритмів для виконання функції розпізнавання, так як у кінцевому результаті приріст значення ймовірності розпізнавання обличчя може бути й не таким вже значним.

Пошук і знаходження людських облич на зображеннях (face detection) є першим важливим кроком у вирішенні проблеми розпізнавання облич у відеопотоках. Задача виявлення людських облич використовується в комп'ютерних системах для вилучення зображень людей з отриманого відеопотоку і подальшого розпізнавання облич або визначення їх положення в тривимірному просторі [19].

Перш ніж виконати будь-який процес виявлення та розпізнавання людини, необхідно виявити обличчя людини на отриманому цифровому зображенні. Відомі інструменти для виявлення та вилучення облич завжди повинні працювати на вхідних цифрових зображеннях, отриманих в реальному часі. Процес виявлення, вилучення та розпізнавання облич можна розділити на наступні етапи, беручи до уваги початкові етапи отримання цифрового зображення:

- отримання необхідного цифрового зображення з відеопотоку;
- пошук та виділення облич з отриманих цифрових зображень;
- етап ідентифікації обличчя шляхом виділення ознак, важливих для подальшого розпізнавання людських облич.

Розглянуті основні етапи пошуку, виділення та розпізнавання людських облич можуть бути виконані різними способами. Наразі існує широкий спектр відомих методів пошуку, вилучення та розпізнавання осіб з цифрових зображень облич (наприклад, з використанням перетворень зображень на основі певних емпіричних даних, вилучення та аналізу унікальних ознак людського обличчя, таких як очі, ніс, рот, за контуром об'єкта зображення, за симетрією частин зображення та інші підходи) [4, 22, 23].

Відомі способи ідентифікації людини за обличчям, що включають перевірку наявності людини на зображенні, отриманому за допомогою відеокамери, визначення положення очей на зображенні та співвіднесення положення інших ознак обличчя з очима, визначення характерних ознак точок обличчя людини в момент підтвердження, обчислення ознак схожості з еталоном та порівняння їх з пороговим значенням, порівняння їх з еталоном

(Патент США N5164992) [22]. Недоліком даного методу є складність у проведенні пошуку очей людини та низька точність розпізнавання.

Відомі методи виявлення та розпізнавання облич на цифрових зображеннях облич людини [23]. Основною метою даного методу є виконання завдання оптимізації процесу розпізнавання облич на цифрових зображеннях облич людини, забезпечення умов для отримання та розпізнавання облич на зображеннях з різних ракурсів, які можуть відрізнятися від фронтального вигляду. Кінцевий результат від застосування методу отримується шляхом виконання ряду послідовних операцій. Першим етапом процесу є введення отриманого цифрового зображення в комп'ютер; потім комп'ютер масштабує зображення і виконує попередню обробку для пошуку та ідентифікації частин і орієнтирів на зображенні людського обличчя. На основі знайдених і виділених опорних точок обличчя людини (центри і краї очей, рота і центр перенісся) розраховуються кути нахилу і повороту голови людини на цифровому зображенні відповідно до їх взаємного розташування, отримане зображення голови людини повертається в потрібну фронтальну позицію, і на нього послідовно накладається оброблений таким чином шаблон зображення. На наступному етапі обробки зображення створюється тривимірна модель людського обличчя, яка масштабується і накладається на попередньо створену еталонну модель людського обличчя. Модель обличчя людини попередньо повертається відповідно до розрахованих кутів нахилу і повороту голови людини, а на наступному етапі всі точки еталонної моделі з'єднуються з відповідними пікселями вихідного зображення для формування фронтальної проекції вхідного зображення обличчя людини і виконання розпізнавання обличчя. Попереднє створення еталонної моделі є необхідною умовою для цього процесу, а етап з'єднання пікселів вихідного цифрового зображення голови людини з усіма точками згенерованої еталонної моделі виконується шляхом повороту моделі, на яку накладені пікселі вихідного цифрового зображення голови людини на попередньому етапі.

Відомий метод пошуку та розпізнавання людських облич на цифрових зображеннях полягає в послідовному накладанні масок на виділені контури людського обличчя для створення нового контуру зображення обличчя, де після першого накладання маски частина зображення за межами накладеної маски відсікається [14]. Перший піксельний контур створюється з частини цифрового зображення, що залишилася після накладання. На наступному етапі на основі першого піксельного зображення створюється нова маска, яка накладається на контур зображення, що містить вирізану частину. На другому етапі створюється другий контур і нова маска, і отриманий другий контур порівнюється з першим контуром. Цей процес продовжується до тих пір, поки різниця відстаней між отриманими контурами не зменшиться до певного постійного значення. До недоліків цього методу пошуку та розпізнавання людей за цифровими зображеннями облич можна віднести складність використання великої кількості шаблонів, оскільки час, необхідний для виконання процесу порівняння та розпізнавання, значно зростає зі збільшенням кількості згенерованих шаблонів. Це пов'язано з тим, що доводиться виконувати значну кількість операцій, таких як створення нових піксельних контурів і нової маски для накладання, а також їх обробка та порівняння контурів між собою.

Основою розглянутого підходу до розпізнавання людських облич на цифрових зображеннях є послідовність, яка гарантує стабільне і надійне розпізнавання обличчя в умовах масштабу, кута повороту голови людини і зміни освітленості, без необхідності використання додаткового спеціалізованого обладнання. Для вирішення задачі пошуку та виділення людських облич необхідно виконати наступну послідовність дій:

- здійснити пошук обличчя людини на вихідному цифровому зображенні та знайти зіниці людини,
- на основі знайдених координат центру зіниці особи нормалізувати знайдене обличчя на зображенні за масштабом та кутом нахилу голови,
- нормалізує цифрове зображення обличчя за рівнем яскравості,

— виконує низку операцій з обчислення близькості між отриманим зображенням та еталонами, що відповідають різним кутам нахилу голови людини в наявній базі даних.

У свою чергу перераховані етапи розглянутої послідовності по пошуку і виділенню обличчя людини вимагають деяких уточнень.

У розглянутому варіанті процес знаходження обличчя у отриманому цифровому зображенні складається із двох послідовних кроків. На першому кроці виділяється область зображення, що може бути обличчям. На наступному кроці виконується пошук характерних точок обличчя людини. Здійснюється пошук центру перенісся обличчя людини та формується множина ознак обличчя, що ґрунтується на описі пари зіниць очей обличчя людини, вершини перенісся, куточків губ, і є основою для подальшого розпізнавання.

Для цього відбувається пошук невеликої кількості точок у отриманому фрагменті зображення, які можуть бути вершиною носу людини. На цьому кроці пошуку використовується детектор точок. Здійснюється знаходження деякої кількості точок за рівнем яскравості, що можуть бути центрами зіниць очей людини у отриманому зображенні та виконується уточнення точок по виділенню центрів зіниць.

На основі знайдених центрів зіниць і вершин носа відбувається поворот зображення до отримання фронтальної проекції і нормалізація зображення по яскравості. Надалі відбувається обробка виділеного фрагменту зображення для розпізнавання особи. Для виконання подальшого розпізнавання особи у виділеному фрагменті зображення здійснюється порівняння нормалізованого зображення виділеного обличчя людини із еталонними зображеннями людей у наявній базі даних.

Можна запропонувати і інший підхід, у якому поворот виділеного фрагменту зображення із обличчям людини до фронтального подання можна і не виконувати. Але тоді необхідно мати достатню кількість зображень обличчя людини із різними значеннями кутів повороту на нахилу обличчя.

У підсумку основні кроки роботи програми по пошуку й розпізнаванню обличчя людини у відео потоці будуть такими.

Виконати процедуру завантаження зображення із відео-камери спостереження. Здійснити попередню обробку виділеного кадру цифрового зображення. Виконати пошук й виділити у отриманому кадрі зображення об'єкти, що можуть бути обличчями людини. Виконати процедуру вирівнювання по яскравості фрагменту цифрового зображення із виділеним обличчям людини. У виділеному в зображенні об'єкті провести пошук характерних ознак обличчя людини. У виділеному об'єкті зображення сформувати перелік ознак, що характеризують обличчя людини. Виконати процедуру ідентифікації обличчя людини шляхом порівняння виділених у даному фрагменті зображення сформованих ознак обличчя із знайденими у наявній базі обличчів ознаками.

Запропонована послідовність кроків по пошуку й виділенню обличчя людини у отриманому відео зображенні лежить в основі методу по розпізнаванню особи, і деякі із етапів реалізації цього процесу розглянуті нижче. Більшу частину етапів цієї послідовності виділення та розпізнавання особи можна покласти на нейронну мережу, попередньо налаштовану та натреновану на сформованій базі зображень обличчів людей, що підлягають розпізнаванню.

2.2 Нейронні мережі для розпізнавання особи

На теперішній час для розпізнавання особи за зображенням обличчя використовують різні типи нейронних мереж, найбільш поширеними із яких є згорткові нейронні мережі. Існує група згорткових нейронних мереж типів MTCNN, R-CNN, YOLO, які найчастіше використовують для розпізнавання обличчя людини. Розглянемо основні характеристики цих мереж для вибору найбільш прийнятного типу нейронної мережі.

У основі побудови нейронних мереж використовують алгоритми виявлення об'єктів, що зазвичай поділяються на дві категорії у залежності від того, скільки раз одне і те ж саме вхідне зображення обробляється мережею.

Однопрохідні детектори об'єктів роблять лише один прохід через вхідне зображення, щоб оцінити наявність і положення об'єктів на зображенні. Оскільки все зображення обробляється за один прохід, він є ефективним з точки зору обчислень (Додаток В до магістерської роботи).

Однак виявлення об'єктів за один прохід, як правило, менш точне, ніж інші методи, і менш ефективне для виявлення невеликих об'єктів. Такі алгоритми можна використовувати для виявлення об'єктів у реальному часі в умовах обмежених ресурсів.

При двопрохідному виявленні об'єктів вхідне зображення пропускається двічі для оцінки наявності та положення об'єктів: перший прохід генерує кандидати в об'єкти, другий прохід уточнює ці кандидати і робить остаточну оцінку. Цей метод є більш точним, ніж виявлення поодиноких об'єктів, але він також є більш обчислювально інтенсивним.

Мережа MTCNN є нейромережевою моделлю, що створена спеціально для пошуку і виділення обличчя на цифровому зображенні. Мережа складається із трьох досить простих згорткових нейронних мереж, які називаються P-Net, R-Net та O-Net. Крім визначення координат обмежувальної рамки навколо виявленого обличчя, модель також визначає координати особливих точок, таких як очі, ніс і рот [21]. Нижче наведено архітектурну схему вищезгаданої нейронної мережі (рис. 2.1) [21].

Модель MTCNN працює в три етапи. Спочатку зображення потрібно передати моделі; MTCNN будує піраміду зображень, щоб модель могла розпізнавати обличчя різного розміру. Іншими словами, модель використовує кілька копій зображень різного розміру.

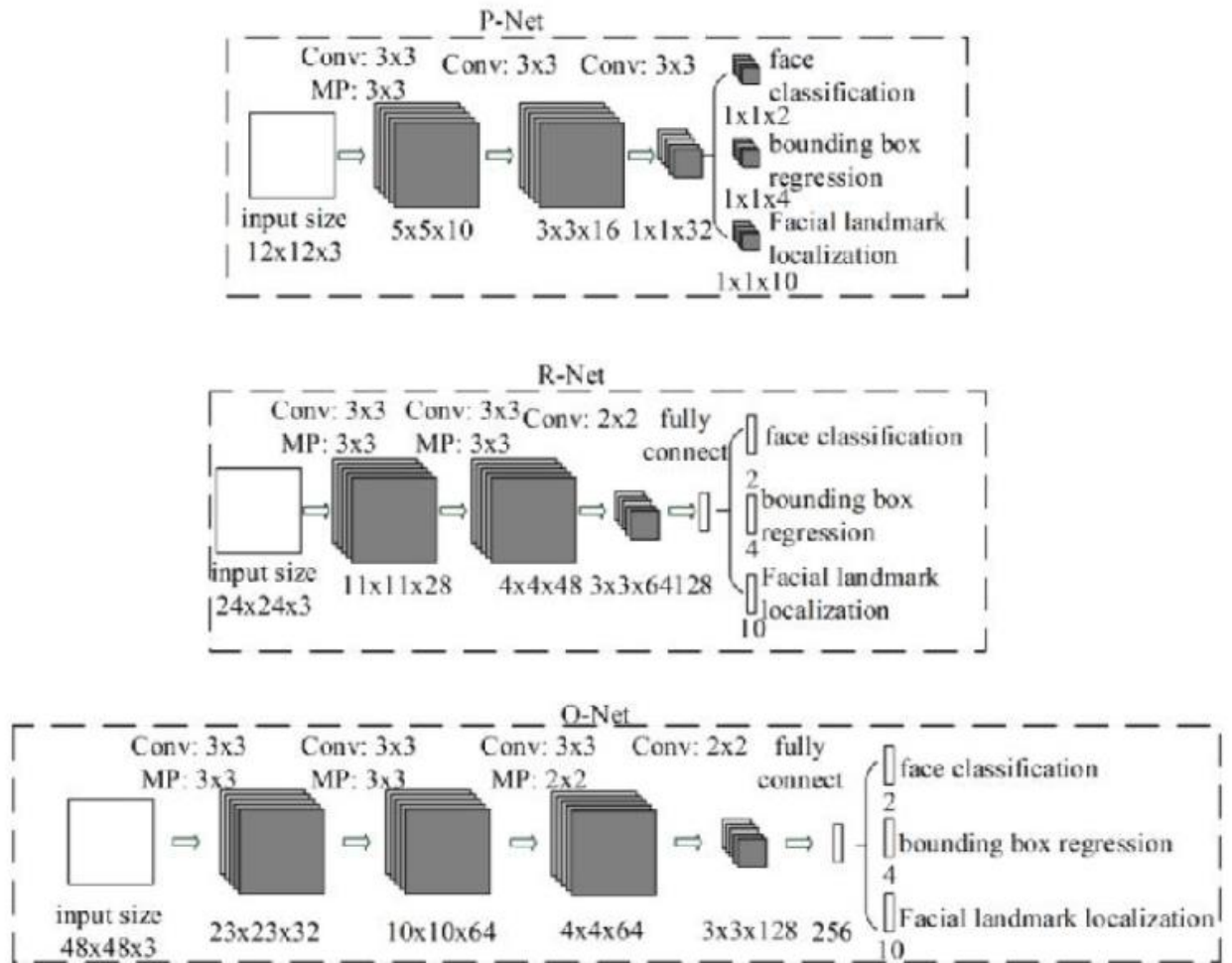


Рисунок 2.1 — Архітектура нейронних мереж MTCNN

Після імпорту зображення в модель створюється кілька масштабованих копій зображення, які передаються до першої нейронної мережі P-Net, де збираються фрагменти зображення із заданими об'єктами. Для кожної копії ядро розміром 12x12 пікселів масштабується до всього зображення зі зміщенням 2. Фрагмент зображення, що містить ядро, обробляється першою нейронною мережею P-Net, яка виробляє координати обмежувальної рамки, якщо на зображенні присутнє обличчя.

Щоб зменшити кількість обмежувальних рамок, використовується алгоритм не-максимального придушення (Non-Maximum Suppression, NMS). В MTCNN алгоритм NMS виконується шляхом попереднього ранжування обмежувальних рамок (і відповідних їм ядер) за достовірністю або оцінкою. Алгоритм NMS застосовується один раз для кожного зображення шкали, а

потім повторно для всіх інших ядер кожної шкали. Це видаляє непотрібні обмежувальні рамки і звужує пошук до одного повного прямокутника на обличчя. Оскільки обмежувальна рамка може бути не квадратною, короткі сторони збільшуються і перетворюються на квадратні. Отримані таким чином координати растра передаються на другий етап.

Після заповнення масив обмежувальної рамки змінюється до розміру 24x24 пікселів і нормалізується так, щоб його значення знаходилося в діапазоні від -1 до 1.

Тепер зображення передаються у наступну нейронну мережу R-Net. Результати роботи мережі R-Net аналогічні результатам роботи P-Net: вони включають в себе координати нових, більш точно сформованих обмежувальних рамок, які відповідають рівню впевненості кожної із цих обмежувальних рамок.

Знову ж таки, граничні області з низьким рівнем достовірності відкидаються, і алгоритм NMS застосовується для усунення інших непотрібних граничних областей. Координати цих нових меж засновані на межах, отриманих за допомогою P-Net, і повинні бути перетворені в стандартизовані координати. Після того, як координати стандартизовано, обмежувальні рамки перетворюються на квадрати і передаються в остаточну нейронну мережу O-Net.

Обмежувальні рамки за межами зображення заповнюються нулями. Потім всі рамки змінюються до розміру 48 x 48 пікселів. Після цього вони імпортується в O-Net. Результати роботи O-Net трохи відрізняється від результатів P-Net і R-Net. Мережа O-Net формує три результати: координати обмежувальних рамок, координати 5 особливих точок у кожній рамці і рівень впевненості кожної із розглянутих рамок.

Модель MTCNN має ряд переваг. Продуктивність моделі є високою завдяки її низькій обчислювальній складності. Модель також починає пошук особливих точок тільки в кінцевій мережі (O-Net) після калібрування координат обмежувальної рамки, що звужує область, в якій модель шукає

особливі точки. Наявність трьох мереж (кожна з декількома шарами) забезпечує більшу точність, оскільки кожна мережа може більш точно калібрувати результати попередньої мережі.

Одним з підходів до виявлення та розпізнавання об'єктів є поєднання регіонів зі згортковими нейронними мережами (Region Based Convolutional Neural Networks: R-CNN). R-CNN — це двоетапний алгоритм виявлення об'єктів. На першому етапі визначається підмножина областей зображення, які можуть містити об'єкти, що підлягають розпізнаванню. На другому етапі проводиться класифікація об'єктів всередині кожної області; для уникнення проблеми вибору великої кількості областей будується R-CNN модель (рис.2.2). Пропонується метод, що використовує вибірковий пошук, для відбору лише 2000 регіонів із зображення.

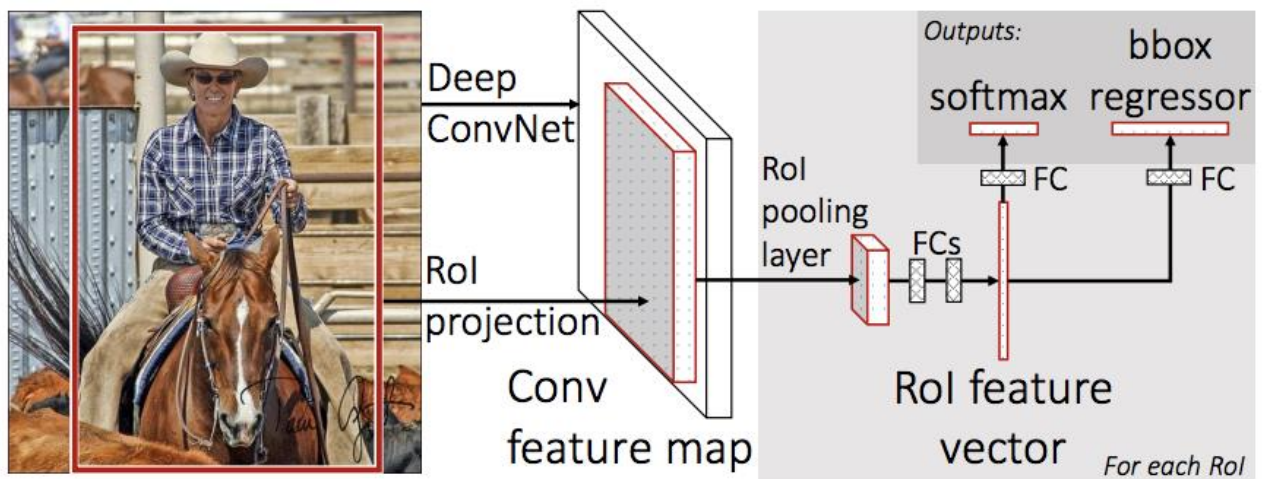


Рисунок 2.2 — Архітектура нейронної мережі R-CNN

Наступна версія Faster R-CNN, включає алгоритм виявлення об'єктів, який усуває потребу в алгоритмах вибіркового пошуку і дозволяє нейронній мережі навчитися самостійно знаходити ознаки регіону [21]. Обидва вищезгадані алгоритми (R-CNN та Fast R-CNN) використовують вибірковий пошук для пошуку регіональних речень. Вибірковий пошук займає багато часу і впливає на продуктивність мережі. Тому Faster R-CNN виключає алгоритм

вибіркового пошуку і вводить алгоритм виявлення об'єктів, який дозволяє мережі вивчати регіональні пропозиції. Як і в Fast R-CNN, зображення подаються на вхід згорткової мережі, яка виконує згортку. Замість того, щоб використовувати алгоритм вибіркового пошуку на карті об'єктів для ідентифікації пропозицій регіонів, інша мережа використовується для прогнозування пропозицій регіонів. Виявлені ставки регіонів перетворюються у вектори фіксованої довжини шаром об'єднання ROI і використовуються для класифікації зображень у запропонованих регіонах та ідентифікації обмежувальних рамок.

Faster R-CNN значно швидша за своїх попередників R-CNN та Fast R-CNN. Тому модель можна використовувати для виявлення об'єктів у реальному часі. На жаль, модель Faster R-CNN все ще потребує значних обчислювальних ресурсів для ефективного навчання, тому через цю особливість по відношенню до обчислювальних ресурсів Faster R-CNN не будемо використовувати у нашій роботі.

YOLO - ще одна група нейромережевих моделей для пошуку об'єктів на зображеннях; як і моделі R-CNN, моделі YOLO еволюціонували послідовно; моделі YOLO зосереджені на швидкому розпізнаванні об'єктів на зображеннях, жертвуючи точністю заради швидкості [24]. Тому моделі YOLO можуть обробляти зображення набагато швидше, ніж моделі R-CNN, і підходять для застосувань, де потрібна швидка обробка зображень, наприклад, для виявлення об'єктів у потоковому відео. Модель YOLO має різні версії, від YOLOv1 до YOLOv8.

YOLO - це однопрохідний детектор, який використовує повністю згорткову нейронну мережу (CNN) для обробки зображень.

You Only Look Once (YOLO) пропонує використання наскрізної нейронної мережі, яка одночасно виконує оцінку ймовірності класів і граничних областей. Це відрізняється від підходу, який застосовували попередні алгоритми виявлення об'єктів, що повторно використовували класифікатори для виявлення. Застосовуючи зовсім інший підхід до виявлення об'єктів, YOLO досягає кращих

результатів і значно перевершує інші алгоритми виявлення об'єктів у реальному часі.

У той час як такі алгоритми, як Faster RCNN, використовують рекомендаційну мережу для визначення областей інтересу і розпізнавання їх окремо, YOLO використовує один повністю пов'язаний шар для всіх прогнозів. Методи, що використовують рекомендаційні мережі, вимагають декількох ітерацій для одного зображення, тоді як YOLO вимагає лише однієї ітерації.

Ще одним покращенням у новій версії YOLO є підтримка блоків з різними масштабами та співвідношенням сторін. Масштаб і співвідношення сторін геоприв'язаних кадрів були змінені, щоб краще відповідати розміру і формі виявлених об'єктів. Варіант архітектури мережі YOLO із різними розмірами згорткових шарів приведено на рис. 2.3.

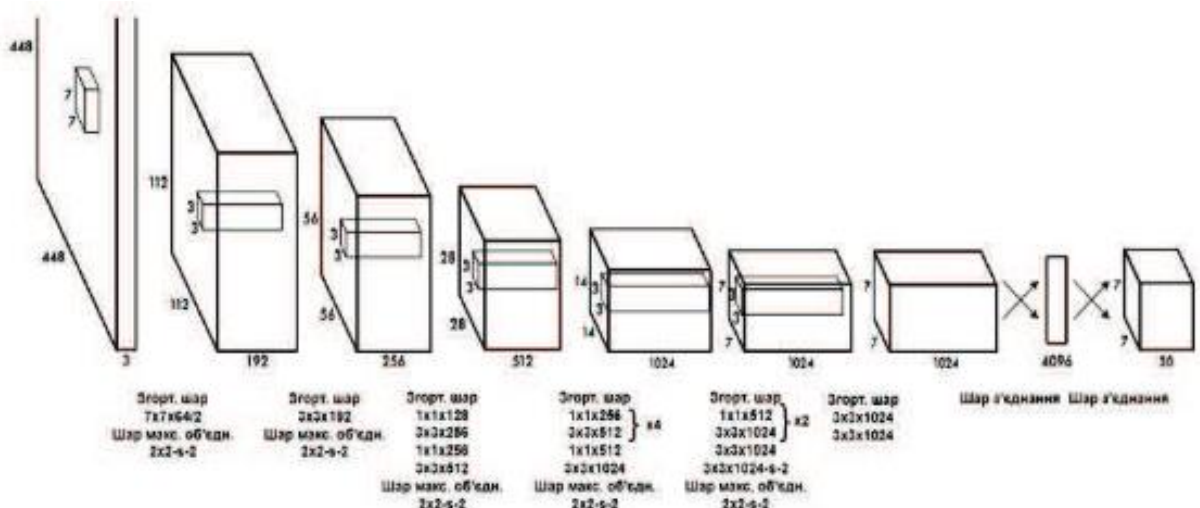


Рисунок 2.3 — Архітектура мережі YOLO

У мережі YOLOv3 також введено концепцію «пірамідальних мереж ознак» (FPN) для архітектури CNN, яка застосовується для виявлення об'єктів у різних масштабах. Алгоритм FPN будує піраміди із карт ознак, де кожен рівень піраміди застосовується для виявлення об'єктів, що подані у різних масштабах. Такий підхід сприяє підвищенню ефективності пошуку невеличких об'єктів та дозволяє виділяти об'єкти із різним масштабом.

У версії YOLOv5 використовується більш складна архітектура, засновану на мережевій архітектурі EfficientNet, для досягнення вищої точності та кращого узагальнення для ширшого діапазону категорій об'єктів.

YOLO v5 також вводить поняття просторової піраміди злиття (SPP), тип шару злиття, що використовується для зменшення просторової роздільної здатності карт об'єктів SPP використовується для того, щоб моделі могли бачити об'єкти в різних масштабах і підвищити ефективність виявлення малих об'єктів. Моделі YOLO працюють за наступними принципами

Модель YOLO працює за наступним принципом: YOLO розбиває вхідне зображення на сітки $S \times S$ і бере m обмежувальних рамок у кожній сітці. Для кожного з них мережа виводить ймовірність класу та зміщення. Обмеження з ймовірністю класу вище порогового значення вибираються і використовуються для пошуку об'єктів на зображенні. Кожна комірка сітки відповідає за пошук лише одного об'єкта. Кожна комірка визначає фіксовану кількість обмежувальних рамок.

Архітектура мережі YOLO складається з декількох послідовних блоків, які працюють послідовно один за одним та дозволяють здійснювати пошук об'єктів на зображенні або ж відео в масштабі реального часу. Основні блоки архітектури YOLO є такі.

Згортковий шар: призначений для зменшення розміру зображення та вилучення з нього об'єктних ознак. Шар Maxpooling: призначений для зменшення розміру зображення та покращення інваріантності до зміщення; блок Convolutional Layer + Batch Normalisation призначений для подальшого вилучення ознак об'єкта та стабілізації градієнта під час навчання.

Нормалізація + Leaky ReLU призначений для додавання нелінійних ознак та покращення розпізнавання об'єктів. Шар повного злиття призначений для зменшення вихідних значень попереднього шару до кінцевих розмірів для отримання вектора ознак. Шар векторизації призначений для перетворення вектора ознак у вектор маркування, що містить інформацію про положення та клас об'єкта.

Шар зв'язування призначений для об'єднання отриманих векторів маркування в єдиний вектор, що містить інформацію про всі об'єкти на зображенні.

Далі йдуть п'ять ітераційних блоків, кожен з яких складається з власного детектора об'єктів, що оперує вхідними даними з переднього кінця мережі.

Кожен детектор об'єктів використовує координати обмежувальних рамок і якірний фільтр для оцінки ймовірності класу кожної з них. Якщо ймовірність перевищує порогове значення, кадр асоціюється з даними класу; YOLO використовує функцію втрати помилок за сумою квадратів, яку легко оптимізувати. Однак ця функція надає рівну вагу завданням класифікації та локалізації.

YOLO набагато швидший за інші алгоритми виявлення об'єктів. Одним з обмежень алгоритму YOLO є те, що він не може впоратися з дрібними об'єктами на зображенні, наприклад, зграю птахів може бути важко виявити. Це пов'язано з просторовими обмеженнями алгоритму.

Виходячи із розглянутого, вибираємо згорткову нейронну мережу YOLO, яка має досить високу швидкість у обробці зображень та дозволяє оперативно проводити пошук та розпізнавання особи у отриманому відеопотоці.

2.3 Навчання нейронної мережі

Розпізнавання особи у відео потоці складається із двох процесів:

- власне процес розпізнавання із етапів отримання відеозображення, пошуку і виділення особи із використанням попередньо навченої згорткової нейронної мережі (основний процес роботи по розпізнаванню),
- і окремо процес навчання згорткової нейронної мережі (допоміжний процес).

Процес навчання нейронної мережі складається із таких кроків, як підготовка даних у вигляді бази даних із різними типами обличч людей, вибір

архітектури згорткової мережі, її тренування і тестування для подальшого використання.

Спочатку необхідно підготувати навчальний набір даних, що складається із зображення та маски, з яких попередньо відбираються відповідні мітки або об'єкти. Потім обирається архітектура нейронної мережі. Після того, як архітектура мережі обрана, її потрібно навчити, використовуючи навчальний набір даних. Навчання мережі передбачає оновлення ваг моделі за допомогою градієнтного спуску для мінімізації функції втрат.

Процес навчання глибокої нейронної мережі вимагає великої кількості підготовчих даних і високої обчислювальної потужності. Нейронні мережі навчаються шляхом ітеративної оптимізації ваг нейронів для зменшення помилок прогнозування. Цей процес вимагає значного часу і ресурсів, але після навчання нейронні мережі можна використовувати для швидкого та ефективного розпізнавання і прогнозування.

Після того, як мережа навчена, її тестують на іншому наборі даних, щоб оцінити її продуктивність. Якщо тест пройдено успішно, мережу можна використовувати для вилучення об'єктів з нового зображення. При використанні згорткових нейронних мереж для успішного виконання завдання потрібно виконати низку важливих кроків, зокрема згортку, об'єднання, згладжування та повне підключення.

Згортання виконується на зображенні для ідентифікації конкретних елементів зображення. Згортка може розмити, підвищити різкість, виявити краї та зменшити шум на зображенні, допомагаючи машині вивчити певні характеристики зображення.

Отримане зображення може бути занадто великим і його потрібно зменшити. Злиття виконується, щоб зменшити зображення без втрати будь-яких особливостей або шаблонів.

Згладжування перетворює двовимірну матрицю ознак у вектор ознак, який можна подати в нейронну мережу або класифікатор. Повне злиття - це

процес простого завантаження модифікованого зображення в нейронну мережу.

Суть згорткової нейронної мережі полягає в інтерпретації від конкретних ознак зображення до більш загальних і абстрактних деталей, інтерпретації до ще більш абстрактних деталей і наступних етапів, що ведуть до генерації і відбору концепцій на найвищому рівні. У цьому процесі нейронна мережа налаштовується, вибудовуючи ієрархію найсуттєвіших абстрактних ознак, відкидаючи несуттєві деталі та підкреслюючи суттєві ознаки [18].

Ядро згортки — це фактично набір заздалегідь визначених вагових коефіцієнтів. В результаті цього процесу також створюється зображення, яке надалі називається картою ознак. Залежно від коефіцієнтів обраного ядра згортки, карта ознак підкреслює певні особливості вихідного вхідного зображення.

Для вилучення найбільш повних і якісних ознак оригінального вхідного зображення використовується кілька різних ядер згортки, в результаті чого на виході шару згортки отримуємо кілька карт ознак. Ядро згортки - це фільтр або вікно, яке переміщується по всій попередній карті, щоб знайти специфічні особливості об'єкта зображення.

Скалярний результат процесу згортки є входом для функції активації. Раніше для цього традиційно використовували дотичні до гіперболи та сигмоїди. На сьогоднішній день функція ReLU є більш поширеною. Це пов'язано з тим, що її простота прискорює процес навчання мережі і спрощує обчислення. Функція активації ReLU (Rectified Linear Unit) використовується до результатів внутрішнього підсумовування. Формула активації ReLU виглядає таким чином:

$$\text{ReLU}(x) = \max(0, x), \quad (2.1)$$

де x - вхідна величина.

Навчання нейронної мережі — це, по суті, вибір моделі, що мінімізує витрати, з множини допустимих моделей (у байєсівських системах

визначається розподіл на множині допустимих моделей). Для навчання нейромережевих моделей існує низка алгоритмів, більшість з яких можна розглядати як пряме застосування теорії оптимізації та статистичного оцінювання.

У більшості випадків використовується певна форма градієнтного спуску, а для обчислення істинного градієнта застосовується зворотне поширення. Це просто робиться шляхом взяття похідних функції витрат по відношенню до параметрів мережі та зміни цих параметрів у напрямку, що відповідає градієнту.

Існують такі функції активації, як сигмоїдна, гіперболічний тангенс та випрямлена лінійна функція активації (rectified linear unit, ReLU). Сигмоїдні функції активації належать до класу неперервних функцій, які приймають на вхід будь-яке дійсне число і виводять дійсне число в діапазоні від 0 до 1. Історично сигмоїд широко використовувався, оскільки його вихід можна добре інтерпретувати як рівень активації нейрона (рис. 2.4).

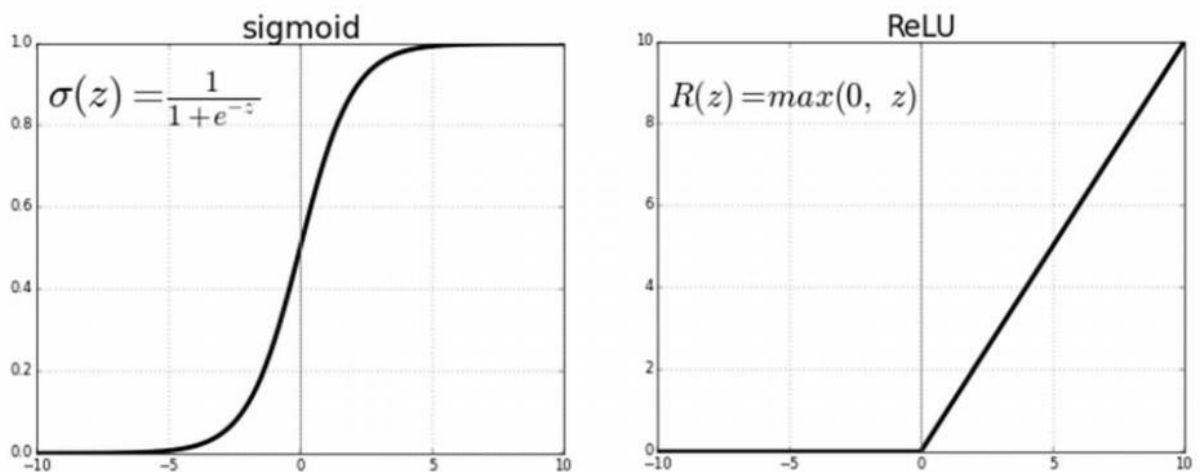


Рисунок 2.4 — Функції активації мережі

Вкрай небажаною властивістю сигмоїда є те, що коли функція насичується з одного боку (0 або 1), градієнт цієї частини близький до нуля. У процесі зворотного поширення помилки цей (локальний) градієнт множиться на загальний градієнт. Тому, якщо локальний градієнт занадто малий, він фактично перекриває загальний градієнт. В результаті сигнал проходить через

нейрон, щоб досягти ваг, і рідко досягає даних рекурсивно.

Гіперболічна дотична функція активації має симетричну функцію активації і сходиться швидше, ніж стандартна логістична функція. Функція має прості похідні, які можна обчислити за їх значеннями, що економить обчислення.

Відомо, що нейронні мережі здатні апроксимувати будь-яку складну функцію за умови наявності достатньої кількості шарів і нелінійної функції активації. Сигмоїдальні та тангенціальні функції активації є нелінійними, але призводять до проблем спаду та розширення градієнта. Однак простішим варіантом є використання випрямлених лінійних одиниць (ReLU). Графік функції активації ReLU наведено на рис. 2.4.

Використання функції ReLU призводить до проріджування ваг. В той час як обчислення сигмоїдного та гіперболічного тангенса вимагає ресурсоемних операцій, ReLU може бути реалізовано за допомогою простої операції порогового значення, яка встановлює матрицю активації в нуль.

З іншого боку, ReLU не завжди достатньо надійний і може давати збої під час навчання. Наприклад, якщо градієнт, що проходить через ReLU, великий, ваги оновлюються і цей нейрон більше ніколи не активується. В такому випадку, з цього моменту градієнт, що проходить через цей нейрон, завжди буде дорівнювати нулю. Ця проблема вирішується вибором відповідної швидкості навчання.

Для навчання згорткових нейронних мереж використовуються алгоритми зворотного поширення. Цей метод навчання багат шарових нейронних мереж називається узагальненим дельта-правилом. Цей алгоритм є першим і основним алгоритмом, який можна практично використовувати для навчання багат шарових нейронних мереж. Вага прихованого нейрона змінюється прямо пропорційно похибці нейронів, до яких цей нейрон підключений. Тому ваги всіх міжшарових зв'язків можуть бути скориговані шляхом зворотного розповсюдження помилок по мережі. У цьому випадку значення функції помилки зменшується і мережа навчається.

Як метод мінімізації помилки використовується метод градієнтного спуску. Суть цього методу полягає в тому, щоб знайти мінімальне (або максимальне) значення функції, рухаючись вздовж вектору градієнта. Для знаходження мінімального значення необхідно рухатися в напрямку, протилежному градієнту [26].

Алгоритм поширення помилки складається з наступних кроків: безпосереднє поширення сигналу через мережу і розрахунок стану нейрона; розрахунок значення помилки Δn для вихідного шару; потім послідовний розрахунок Δn для всіх прихованих шарів від хвоста до вершини і зворотнє поширення для оновлення ваг мережі відповідно до раніше розрахованих помилок Δn .

Для оцінки точності роботи нейронної мережі по розпізнаванню об'єктів використовується функція втрат (loss function), що характеризує якість роботи нейронної мережі. Такою функцією втрат є категоріальна функція ентропійних втрат (CategoricalCross-EntropyLoss function, формула 2.2), у якій y^* є фактичною відповіддю нейронної мережі, а значення y є бажаним.

$$H(y, y^*) = - \sum y_i \log y_i^*. \quad (2.2)$$

Для оцінки якості налаштування нейронної мережі використовуємо критерій оптимальності, в якості якого використовуємо квадратичний критерій ξ [18]

$$\xi = \frac{1}{2} \sum_{s=1}^S \sum_{p=1}^M \left(y_p^s - \text{HM}_p^s(x^s) \right)^2 \quad (2.3)$$

де s – номер екземпляра;

S – кількість екземплярів;

p – номер вихідної ознаки;

M – кількість вихідних (цільових) ознак;

y^s – набір значень вихідних ознак, зіставлений s -му екземпляру;

HM_p^s – розраховане значення на p -му виходу нейронної мережі при подані на її входи набору значень вхідних ознак x^s ;

x^s – набір вхідних ознак.

Згорткові нейронні мережі використовують стратегію розподілу ваг, яка призводить до значного зменшення кількості параметрів, що підлягають вивченню. Наявність більших розмірів нейронів-приймачів у послідовних згорткових шарах також призводить до трансляційної інваріантності, пов'язаної з наявністю зв'язаних шарів. Результат прямого і зворотного розповсюдження залежить від того, який шар розповсюджується [31].

Перераховані етапи налаштування нейронної мережі дозволять підготувати мережу для виконання операцій пошуку і розпізнавання особи у відео потоці.

У цьому розділі роботи розглянуто низку задач, пов'язаних із пошуком та розпізнаванням особи за зображенням обличчя людини із використанням нейронних мереж. Основний підхід до пошуку й розпізнавання людини у отриманому відеопотоці полягає у пошуку людини шляхом виділення точок, що є характерними для зображення обличчя людини. Для пошуку і виявлення зображення людини та визначення особи використано згорткову нейронну мережу із глибоким навчанням. У розділі роботи також сформована послідовність обробки відео зображення для пошуку і подальшого розпізнавання особи у вибраній сцені. Запропонований у магістерській роботі підхід до виділення зони у відео зображенні із людьми та використання згорткової нейронної мережі дозволяє реалізувати процес обробки відеозображень для пошуку і розпізнавання особи за зображенням обличчя людини.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ПОШУКУ ТА РОЗПІЗНАВАННЯ ОСОБИ

У даному розділі магістерської роботи розроблені програмні засоби для виконання дій по здійсненню пошуку і подальшому розпізнаванню особи в отриманій відеопослідовності кадрів із застосуванням згорткової нейронної мережі.

3.1 Формування архітектури програми розпізнавання особи

Процес пошуку, виділення та розпізнавання осіб в зоні відеоспостереження пропонується проводити в кілька етапів. На цих етапах створюється загальна програмна архітектура для роботи систем відеоспостереження, які виконують пошук і ідентифікацію людей. На першому етапі програми виконується вибір області отриманого зображення, в якій шукається зображення обличчя людини. Наступним кроком буде використання вибраного та обробленого зображення обличчя людини для подальшої ідентифікації обличчя за допомогою згорткової нейронної мережі.

Програмний пакет, який використовується для ідентифікації та ідентифікації людини, працює в режимі реального часу — зображення знімається кадр за кадром з встановленої камери відеоспостереження, обробляється в запропонованій серії операцій, а результати програми відтворюються на екрані монітора спостерігача. Перед запуском програми потрібно змонтувати файли з бібліотеки OpenCV [33], які необхідні для роботи процесів, інтерпретованих за допомогою мови Python. У роботі програми також використовується згорткова нейронна мережа YOLOv8 і база даних для початкової настройки і навчання нейронної мережі.

Щоб ця система могла працювати над ідентифікацією людей, необхідна група модулів, таких як вибір відеокadrів із отриманого потоку даних, Перевірка вибраних фрагментів зображення та розпізнавання облич. Ці групи модулів є основою програми, кожна з яких активна лише після успішного

завершення попереднього модуля. Від введення даних, введених в систему, до заключного етапу виведення результату ідентифікації.

Група модулів збору зображень відповідає за здатність системи витягувати кадри з відеопотоків для розпізнавання. Він може мати як інтерфейс для вказівки шляху зображення, так і можливість роботи з відеокамерою. Наступним елементом проекрованої системи є деталізація обраного кадру зображення з використанням результатів попередніх етапів роботи. Основне завдання цієї групи модулів-створити вийшло зображення і виділити в ньому всі присутні особи людей. Розпізнавання обличчя на основі зображення обличчя здійснюється шляхом пошуку обличчя потрібної людини за допомогою нейронної мережі за допомогою існуючої бази даних.

Програма запускається з вказівки шляху до файлу для вибору кадру з відеопотоку (модуль CaptureIm). І завантаживши вибраний кадр, приступаємо до роботи над ним. Вибрана частина нормалізованого зображення (модуль NormalIm). Тому підготовлені фрагменти зображення відправляються в згорткову нейронну мережу (модуль RecognIm) для розпізнавання. Процес розпізнавання виконується з використанням попередньо підготовлених даних про обличчя людини (ImBaseHuman). Результати розпізнавання осіб відображаються на екрані монітора для спостерігача. Якщо під час процесу розпізнавання вибраний об'єкт не був ідентифікований через його відсутність у базі даних, з'являється повідомлення про те, що ця особа відсутня в базі даних. Згорткова нейронна мережа, що виконує процес ідентифікації, проходить етапи попереднього налаштування та навчання за допомогою модуля CNNLearn.

Для створення програмного забезпечення використовувалося кілька готових файлів з бібліотеки з відкритим вихідним кодом, з яких була обрана інформація, необхідна для роботи цього програмного пакету. Програмний пакет, створений для пошуку і подальшого розпізнавання людських образів, був створений з використанням мови програмування Python і засобів бібліотек NumPy, OpenCV і Dlib.

Архітектура програми розпізнавання осіб у відеопотоці складається з наступних модулів: модуля захоплення кадру зображення з використанням бібліотеки OpenCV, модуль попередньої обробки зображення, модуль виведення зображення й модулі розпізнавання особи із використанням нейронної мережі та модуля навчання нейронної мережі. Перерахований склад модулів для виділення і розпізнавання особи приведено у додатку Г до магістерської роботи.

Модуль вибору відеокадру забезпечує можливість отримання кадру зображення з периферійного пристрою, підключеного до системи. Далі йде робочий цикл виконання програми — кадр зображення береться з камери і виводиться у вікні. Вибираючи окремі відеокадри, система використовує нейронну мережу для подальшого розпізнавання облич та обробки результатів. Захоплення відеокадру реалізовано за допомогою пакета `opencv-python` та реалізовано з бібліотеки `python OpenCV`. Зображення OpenCV зберігаються в багатоканальному масиві типу `numpy` за допомогою модифікованої кольорової системи BGR. Для подальшого використання обраного зображення модулем нейронної мережі досить виконати зворотне перетворення колірної схеми і розширити масив кількома додатковими вимірами.

Наступний модуль виконує попередню обробку зображення для подальшого пошуку і розпізнавання особи у отриманому кадрі відеопотоку.

На модуль виведення зображення покладається завдання виведення розпізнаного зображення особи. Якщо вхідна черга не заповнена, наступний кадр зчитується з поступаючого відеопотоку та поміщається у чергу. Кадри відеопотоку зчитуються й поміщаються до сформованої черги із відповідними порядковими номерами кадрів, тобто фактично об'єкт із списку Python поміщається в таким чином сформовану чергу. Потім обирається наступний кадр із вхідної черги, обробляється та поміщається у чергу для виведення із своїм сформованим номером кадру.

Для функціонування створеного програмного продукту була встановлена модель нейронної мережі YOLOv8. Так як більшість створених програм використовують згорткову нейронну мережу по декілька разів із різними областями аналізованого зображення, то YOLOv8 використовує її всього один раз до усього зображення. Мережа YOLOv8 ділить отримане зображення на сітку й передбачає розташування подібних об'єктів для кожної ділянки. Мережа YOLOv8 працює значно швидше за відому мережу типу R-CNN, що є суттєвим показником для обробки інформації в режимі реального часу на комп'ютерному пристрої. Дана мережа також має багато різних реалізацій у вигляді файлів, одним із яких Keras. Концепція використання процесу розпізнавання об'єктів із застосуванням YOLOv8 передбачає перехід до завдання у вигляді регресійної форми. Для виявлення об'єктів нейронна мережа YOLOv8 задіює однопрохідний алгоритм глибокого навчання. Принцип роботи мережі YOLOv8 полягає у тому, що вибране зображення розділяється на n-ну кількість однакових частин (фрагментів), у кожному із яких виділяється центр осередку зображення. Кожен із осередків передбачає фіксовану кількість обмежуючих рамок зі значенням достовірності розпізнавання (confidence score). Обмежувальні рамки формуються на базі п'яти складових: враховуються координати по осі абсцис й ординат, вводиться висота й ширина обмежувальної рамки, і також значення достовірності розпізнавання. Для вибору найкращого значення використовується підхід IoU, що здійснює перетин над об'єднанням. Крім того, у мережі YOLOv8 включена опція немаксимального придушення для видалення можливих зайвих рамок. Також алгоритм доповнений пакетною нормалізацією (batch normalization) разом із згортковими шарами для зменшення імовірності перенавчання мережі та підвищення точності її роботи.

3.2 Розробка програми розпізнавання особи

У попередніх підрозділах роботи була запропонована послідовність пошуку і розпізнавання особи, основними складовими якої є виділення кадру

відеозображення, попередня його обробка, пошук зон із обличчям людини та розпізнавання особи у виділеному фрагменті зображення. Для виділення кадру зображення можна скористатися відомими підходами із бібліотеки OpenCV. Пошук зони із зображенням обличчя людини можна здійснити різними способами. Більш швидкодіючим є використання детектора Віоли-Джонса. Його і використаємо у нашому програмному продукті. Із його допомогою локалізуємо області зображення, де є обличчя людини, в результаті чого визначається фрагмент зображення для подальшого розпізнавання із застосуванням згорткової нейронної мережі.

Результатом розробки програми є використання детектора ознак обличчя людини, який використовується у режимі реального часу, та завершальна функція роботи програмного продукту у вигляді функціоналу по розпізнаванню особи. Для функціонування програми розпізнавання особи, перш за все, необхідно підключитися до бібліотеки типу OpenCV із відкритим вихідним кодом та установити необхідні для подальшого формування програми компоненти із цієї бібліотеки — cv2, numpy, dlib та face_recognition. Для роботи із нейронною мережею та внесення необхідних нейронів приміняємо засіб NumPy, у якому за замовчуванням уже знаходяться вихідні дані та програмний код цієї мережі. Для роботи програми використовуємо також засоби із обчислювальної бібліотеки Python, у якій програмно реалізовано значна кількість необхідних для нашої розробки математичних операцій.

Лістинг 3.1 — Фрагмент коду завантаження бібліотек

```
from imutils.video import VideoStream
import dlib
import cv2
import face_recognition
import numpy as np
```

Для функціонування програми підключаємо необхідний детектор, що виділяє обличчя людини за попередньо налаштованими ознаками по ключових точках.

Лістинг 3.2— Фрагмент коду підключення детекторів

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(
    shape_predictor_face_landmarks.dat)
```

Наступним кроком є підключення відео-потoku із камери спостереження. Формуємо функціонал ініціалізації камери, виконуємо зчитування кадрів із неї, а також створення черги кадрів для пошуку і розпізнавання людей. Цей функціонал відноситься до класу опису потоку `CameraStreamReaderThread`. Використовуємо метод `initialize_camera`, який створює об'єкт класу `VideoCapture` (із відкритої бібліотеки `OpenCV`) та передає індекс підключеної камери. Якщо на цьому етапі камера була успішно ініціалізованою, тоді вказується її роздільна здатність. Ініціалізація камери виконується методом `run` та надсилається сигнал із статусом цієї ініціалізації. Після успішної ініціалізації камери запускається цикл послідового зчитування кадрів із неї та формування сигналів із кожним успішно зчитаним кадром. Після запуску потоку для знаходження людини метод додає кожен кадр у чергу і очікує на його обробку. По завершенню процесу пошуку і розпізнавання особи виконується звільнення ресурсів камер. Фрагмент програми по ініціалізації камери та її функціонуванню приведено у лістингу 3.3, повний код приведено у Додатку Д до магістерської роботи.

Лістинг 3.3 — Фрагмент коду ініціалізації камери

```
def __initialize_camera(self):
    self.video_capture = cv.VideoCapture(self.camera_index)
```

```

        if not self.video_capture.isOpened(): return False
    self.video_capture.set(cv.CAP_PROP_FRAME_WIDTH,
        self.camera_resolution[0])
    self.video_capture.set(cv.CAP_PROP_FRAME_HEIGHT,
        if is_successful_camera_frame_read:
            self.camera_frame_read.emit(camera_frame)
        if self.is_person_location_detection_running:
            self.camera_frames_to_process.put(camera_frame)
        self.camera_frames_to_process.join()
    self.video_capture.release()

```

Виконується попередня обробка отриманого кадру зображення та пошуку у ньому зони із обличчям людини. Для виявленого фрагменту зображення із обличчям людини формуємо координати контрольних точок зображення обличчя людини та здійснюємо їх побудову у виділеному зображенні (див. Додаток Д до магістерської роботи).

Надалі приступаємо до процесу ідентифікації особи. Для цього формуються координати вершин прямокутника на даному фрагменті зображення із виявленим зображенням обличчя людини та надсилається на згорткову нейронну мережу, яка використана при створенні даної програми.

Наступним етапом є послідовність дій по ініціалізації навченої моделі згорткової нейронної мережі для виконання процесу розпізнавання особи. Метод `initialize_detection_model` формує об'єкт класу `dnn_DetectionModel` (із тієї ж бібліотеки `OpenCV`), та передає їй шлях до файлу із навченими вагами згорткової нейронної мережі, а також шлях до файлу із конфігурацією моделі. Після цього задаються параметри, щоб модель мережі працювала із використанням графічного процесора із підтримкою `CUDA`. Наступним етапом є підготовка параметрів вхідного зображення для обробки мережею `YOLO`. Першим вводиться множник для значень пікселів, який дорівнює $1/255$, так як мережа `YOLO` використовує нормалізовані значення пікселів.

Друге значенням вказується розмір зображення для нейронної мережі, який дорівнює 416×416 відповідно до розміру навченої моделі.

Надалі використовуємо функціонал для розпізнавання об'єктів у отриманому кадрі. Метод `detect_camera_frame_objects` використовує виділений об'єкт для навченої нейронної мережі для виконання процесу детектування і розпізнавання людини у отриманому кадрі. На виході роботи мережі отримуємо ідентифікатори класів, значення рівня впевненості у розпізаному фрагменті зображення та обмежувальні рамки для розпізнаних об'єктів. Метод `detect` виділяє тільки ті зображення особи, значення рівня впевненості яких вищі за визначене порогове, а зайві обмежувальні рамки відкидаються із використанням порогового значення алгоритмом `non-maximum suppression`. Визначення швидкості роботи нейронної мережі визначається методом `measure_fps`. Швидкість роботи нейронної мережі у FPS обчислюється як одиниця, що поділена на знайдену різницю часу кінця та початку процесу розпізнавання. Ця різниця часу кінця і початку процесу розпізнавання визначається як дійсне значення у секундах (див. Додаток Д до магістерської роботи).

По завершенню роботи програми виконується процес виведення обробленого зображення із позначенням виявленого обличчя людини.

Повний лістинг цієї частини розробленої програми приведено у Додатку Д магістерської роботи.

3.3 Налаштування нейронної мережі

Виконуємо імпортування необхідних бібліотек для налаштування нейронної мережі.

Лістинг 3.4 — Фрагмент коду завантаження бібліотек

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np from sklearn.metrics
```

```
import accuracy_score
import numpy as np
```

Обчислюємо значення сигмоїдної функції та виділяємо її максимальне значення. Обчислюємо значення функції втрат, як різницю між прогнозованим та реальним значенням з урахуванням Л1 регуляризації. Для цього попередньо обраховуємо параметри `predicted`, який є прогнозованим значенням, що вирахований із масиву з 4 значень, та параметра `actual`, що обраховується із реальних значень із масиву із 4 значень. У результаті цих дій отримуємо середнє значення функції втрат.

Лістинг 3.5 — Фрагмент коду обчислення функції втрат

```
def cost_function(predicted, actual):
    return np.mean(np.abs(predicted - actual))
```

Наступним кроком є обчислення функції G для знаходження відношення між попередньою функцією та прогнозованим значенням ознак. Тут обчислюються параметри `previous_function` як попередня функція та `predicted_features` як прогнозоване значення ознак з урахуванням ядра `kernel`.

Надалі виконуємо обчислення функції g для знаходження відношення між значенням функції втрат J та ядром `kernel` через суму G та з урахуванням значення ознак.

Для знаходження відношення між значенням функції втрат J та значеннями ознак через обчислену суму G та ядра `kernel` обчислюємо значення функції h (див. Додаток Е до магістерської роботи).

Обчислюємо функцію оновлення ваг шляхом знаходження різниці попередніх ваг `previous_weights` та значення функції g .

Лістинг 3.6 — Фрагмент коду обчислення функції оновлення ваг

```
def update_weights(previous_weights, g_function_result):
    return previous_weights - g_function_result
```

Надалі обчислюємо функцію оновлення ознак через знаходження різниці попередніх ознак `previous_features` та визначеної функції `h` як параметра `h_previous_features`.

Лістинг 3.7 — Фрагмент коду обчислення функції оновлення ознак

```
def update_features(previous_features, h_function_result):
    return previous_features - h_function_result
```

Формуємо конструктор класу для згорткової нейронної мережі, у якому вказуємо параметри `input_size`, розмір вхідного шару `hidden_layers`, розміри кожного прихованого шару `output_size`, розмір вихідного шару `kernel_size` та розмір ядра конволюції.

Вводимо нові змінні для збереження точності та оцінювання втрат, формуємо структуру для зберігання шарів, ваг та ознак, здійснюємо ініціалізацію ваг і ознак для вхідного шару, прихованих шарів та вихідного шару.

Формуємо функцію `add_layers` для додавання нових прихованих шарів, у якій вказуємо як параметри кількість нових прихованих шарів `num_layers` та розмір кожного нового прихованого шару `layer_size` (див. Додаток Е до магістерської роботи).

Потім створюємо функцію `set_output_layer` для встановлення розмірності міток для вихідного шару `labels_size` та значень розмірності `values_size` для вихідного шару.

Лістинг 3.8 — Фрагмент коду встановлення розмірності міток

```
def set_output_layer_params(self, labels_size, values_size):
    self.layers[-1]["weights"] = np.random.randn(self.hidden_layers[-1],
labels_size)
    self.layers[-1]["labels"] = np.zeros((1, labels_size))
    self.layers[-1]["values"] = np.zeros((1, values_size))
```

Наступним етапом формування програми є створення функції `set_hidden_layer_params` для встановлення розмірів ядра `kernel_size` для прихованого шару та параметру `hidden_features_size` як розміру для входніх ознак для роботи прихованого шару. Для цього як параметр використовується індекс прихованого шару `layer_index`, для якого вже потім установлюються перераховані параметри.

Зберігаємо для порівняння попередні значення втрат (зберігаємо останні три значення втрат), зберігаємо значення точності та отриманих втрат для кожного із оброблюваних зображень, здійснюємо оцінку точності та втрат для виведення, зберігаємо значення точності і втрат для кожного зображення при виведенні, формуємо оцінку точності і втрати для кожного із оброблюваних зображень та виконуємо порівняння обчислених функцій втрат із допустимими значеннями (див. Додаток Е до магістерської роботи).

Перевіряємо критерій зупинки як різницю між останніми трьома втратами, яка має бути не більше `max_difference`) і виводимо повідомлення про завершення процесу на даній епосі.

Лістинг 3.9 — Фрагмент коду перевірки критерію зупинки

```
if len(last_losses) == 3 and max(last_losses) - min(last_losses) <=
max_difference:
    print(f"Training stopped. Last three losses within {max_difference}.")
    break
```

Здійснюємо оцінку точності та виконуємо обрахунок втрат для тестової вибірки, перевіряємо отримання прогнозованих значень. Формуємо результат без використання сигмоїди та припускаємо, що маємо лише один приклад і виводимо зображення у заданому розмірі `image.reshape((28, 28))`.

Виконуємо перевірку точності та виводимо прямокутну зону для кожного прикладу

Формуємо функцію `convolution` для застосування операції конволюції до вхідних ознак `features` із заданим ядром `kernel` та кроком `stride`. Описуємо параметри ядра конволюції (як масив форми `kernel_height, kernel_width`), вхідні ознаки у вигляді масиву форми `(input_height, input_width)` та крок конволюції – `stride`, який за замовчуванням дорівнює 1. Обчислюємо розмір вихідних ознак `output`. Ініціалізуємо результат конволюції, виконуємо сам процес конволюції та у підсумку отримуємо результат конволюції (див. Додаток Е до магістерської роботи).

Для контролю за процесом налаштування мережі вводимо функцію для збереження середніх значень точності та оцінювання втрат для кожної епохи `save_epoch_results`. Обчислення функції втрат виконуємо із використанням дії `cost_function` та здійснюємо збереження значення точності та результати обчислення втрат для кожного зображення. Ці ж дії виконуємо для збереження точності та визначення втрат для кожного зображення у тестовій вибірці. По завершенню роботи зберігаємо результати тестування.

Лістинг 3.10 — Фрагмент коду обчислення значень точності і втрат

```
def save_epoch_results(self, epoch, average_loss, average_accuracy):
    result_string = f"Epoch {epoch + 1}, Average Loss: {average_loss},
Average Accuracy: {average_accuracy}"

    def evaluate_on_test_set(self, X_test, y_test):
        accuracies_per_image = []
        losses_per_image = []
        self.test_accuracies.append(accuracy)
        self.save_test_results(average_loss, average_accuracy)
```

Приведена послідовність налаштування згорткової нейронної мережі дозволили підготувати її для подальшого навчання та тренування на початковому та валідаційному наборах даних. Для налаштування та навчання мережі була створена група навчальних даних та група перевірки.

Як тільки мережева архітектура була побудована, вона була підключена до групи навчальних і перевірочних даних. Також для кращого виконання процесу навчання було створено зручний якір для нейронної мережі. Після налаштування середовища було виконано навчання нейронної мережі на 100 епохах моделі мережі.

Повний лістинг програми приведено у Додатку Е до магістерської роботи.

У даному розділі описано структуру програми, процес підготовки до завантаження відеокадру, налаштування нейронної мережі та послідовність роботи нейронної мережі.

4 ТЕСТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ПОШУКУ ТА РОЗПІЗНАВАННЯ ОСОБИ

У даному розділі роботи виконано тестування програмних засобів пошуку і розпізнавання особи та здійснено перевірку якості роботи програми.

4.1 Навчання і перевірка роботи нейронної мережі

Пошук та розпізнавання об'єктів у цифрових зображеннях на теперішній час залишається однією із надважливих задач у галузі комп'ютерного зору. Незважаючи на великі досягнення, ця галузь постійно вдосконалюється та існує багато напрямків для подальшого поліпшення моделей і методів розпізнавання об'єктів.

Поєднання різних підходів або архітектур нейронних мереж може стати ефективним способом покращення результатів розпізнавання заданих об'єктів. Зокрема, використання поєднання декількох моделей, які об'єднують переваги із кількох підходів, може сформувати вищу точність розпізнавання або поліпшити інший критерій. Пропонується здійснити поєднання детектора пошуку обличчя людини у зоні спостереження із одноетапним детектором, таким як нейрона мережа YOLO. Детектор Віоли-Джонса відомий своєю високою точністю у виділенні зони цифрового зображення, де є обличчя людини, а мережа YOLO забезпечує значну швидкість обробки та розпізнавання особи в реальному часі. Поєднання цих двох підходів може дати певний приріст у результативності розпізнавання особи.

YOLOv8 - це новітнє покоління архітектури мережі для розпізнавання об'єктів, де перегляд і розпізнавання об'єктів виконується за один прохід, і яка значно поліпшена в порівнянні з попередніми версіями. Нейронна мережа YOLOv8 забезпечує вдосконалення механізмів обробки зображень і використання згорткових каскадів для подальшого скорочення часу обробки при збереженні або підвищенні точності виявлення об'єктів в різних масштабах. Ці технічні досягнення роблять YOLOv8 особливо привабливим

для додатків, де потрібна максимальна швидкість обробки в режимі реального часу без шкоди для точності. Процес налаштування моделі YOLOv8 порівняно із попередніми моделями є простішим і швидшим завдяки оптимізації коду та різним вбудованим функціям, які зводять до мінімуму необхідність ручного кодування.

Ще однією важливою перевагою YOLOv8 є те, що вона може ефективно обробляти великі обсяги даних. Це дозволяє використовувати модель у середовищах, де потрібні обчислення, такі як розподілені обчислення та програми на вбудованих пристроях.

Хоча YOLOv8 має високу точність у навчальних наборах даних, її можливості для узагальнення включають різні умови освітлення, кути огляду камер спостереження та фонові об'єкти. Моделям YOLO може бути важко розпізнати невеликі об'єкти в кадрі або об'єкти з низькою контрастністю. Також можливо, що модель YOLOv8 не була достатньо добре тренована або налаштована на деяких специфічних даних, що є в реальному відео. Такі параметри, як розмір навчального набору, кількість періодів або параметри покращення даних, можуть вплинути на кінцевий результат. Якщо у навчальних даних певні класи подані недостатньо, модель YOLOv8 може трохи гірше справлятися завдяки особливостям своєї архітектури і тим методам, які використовуються для оброблення даних про об'єкти. Реальні відео зображення можуть містити різного виду завади, артефакти компресії й інші види дефектів, що впливають на точність розпізнавання. Модель YOLOv8 може бути більш чутливою до таких артефактів завдяки своїй більш простій архітектурі. Таким чином, модель YOLO показує відмінні результати, але з певної причини її ефективність може бути знижена в реальному відео для деяких розмірів об'єктів.

Для тестування і перевірки працездатності розроблюваної програми був обраний персональний комп'ютер із середньою продуктивністю, що включає процесор з мінімальним периферійним обладнанням, вбудовану веб-камеру і засоби зв'язку з ним. Комп'ютер має такі технічні характеристики:

центральний процесор Intel Core i5-12400F із архітектурою Alder Lake, що має 6 ядер та 12 потоків, робоча частота 2,5 ГГц, максимальна частота 4,4 ГГц, оперативна пам'ять 32 GB типу DIMM DDR5, твердотільний накопичувач даних SSD на 1 ТБ, відеокарта NVIDIA RTX 4060 із об'ємом відео-пам'яті 8 ГБ. Для відображення зображень використовується монітор розміром 27" на матриці IPS із роздільною здатністю 1920×1080 типу Full HD та частотою поновлення кадрів 165 Гц.

Перерахованих технічних параметрів комп'ютера було достатньо для перевірки якості розроблюваних програм і дозволило отримати результати аналізу точності створюваної програмної системи.

Нейронна мережа типу YOLOv8 із встановленою архітектурою та вибраною функцією активації ReLU була обрана для розпізнавання облич. Наступним кроком у налаштуванні мережі є запуск масштабу зображення для обробки. Цей параметр нейронної мережі має вирішальне значення для її продуктивності. Також мережа має бути обов'язково навчена. Процес навчання нейронної мережі передбачає коригування ваг, щоб мінімізувати функцію втрат. Навчання нейронних мереж є досить трудомістким процесом, що вимагає значних обчислювальних ресурсів. Щоб виконати навчання, потрібно створити послідовність data.In. У цьому випадку потрібно сфотографувати обличчя під різними кутами огляду. У даній роботі вирішено створити набір даних на основі відкритого коду. Використовуючи такий набір даних, потрібно зберегти його у необхідному форматі та експортувати. Використовуючи згорткову нейронну мережу YOLOv8 для її роботи потрібно працювати в оптимальному форматі подання зображень, яким є формат підготовлених зображень COCO [34].

Наступним суттєвим кроком у створенні навчальних даних для налаштування мережі є розміщення підготовлених зображень відповідно до поставленого завдання згорткової нейронної мережі. Для цього розділяємо сформований набір даних на тренувальну частину й перевірочну (так звану

валідаційну). Дані із цих наборів зображень не повинні між собою перетинатися.

Наступним етапом налаштування мережі є формування опису конфігурації цієї мережі із переліком її гіперпараметрів. До таких параметрів відносяться вхідні дані, розміри зображення для обробки, кількість епох тренування та інші. Відбувається процес навчання моделі мережі для заданої кількості циклів навчання (вказуються як епохи навчання). Як аргументи використовуються вхідні дані та цільові дані, вказується розмір пакету, приводиться кількість циклів навчання моделі у вигляді епох на усій вибірці даних та приводиться режим відображення ходу навчання. Вказується список викликів (callbacks), доля навчальних даних (validation_split), що використовується для перевірки роботи моделі та дані для обрахунку функції втрат та метрик (validation_data) по завершенню кожної епохи навчання, ваги класів (class_weight) при розрахунку функції втрат та ваги екземплярів (sample_weight), що враховується при обрахунку функції втрат. Приводиться епоха, із якої починається процес навчання (initial_epoch), вказується кількість пакетів екземплярів (steps_per_epoch), що обробляються на одній епосі навчання та кількість пакетів (підвибірок) екземплярів (validation_steps) для перевірки моделі мережі перед її зупиненням, а також кількість епох навчання (validation_freq), що виконуються до початку запуску процесу перевірки. Фрагмент опису параметрів нейронної мережі для її налаштування приведено на рис. 4.1.

2	-1	1	19904	models.common.BottleneckCSP	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	3	161152	models.common.BottleneckCSP	[128, 128, 3]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	641792	models.common.BottleneckCSP	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	656896	models.common.SPP	[512, 512, [5, 9, 13]]
9	-1	1	1248768	models.common.BottleneckCSP	[512, 512, 1, False]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	378624	models.common.BottleneckCSP	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]

Рисунок 4.1 — Фрагмент опису параметрів мережі

Використовуємо попередньо навчену модель, засновану на наборі

COCO, для тренування в згортковій нейронній мережі типу YOLOv8. Ці шкали для зображень вже створені й мають певні значення, коли мова йде про пошук досить великої кількості класів елементів. Щоб програма працювала, було використано мережеву архітектуру YOLOv8 і вибрано масштаб, що підготовлений у наборі COCO. Процес навчання нейронних мереж виконується ітеративно. Для цієї мети використано згенерований набір навчальних даних. Процес навчання нейронної мережі складається з покрокового навчання та перевірки кожного етапу. Підготовка даних для навчання приведено на рис.4.2.

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)

train_path = 'C:/Users/acsel/source/repos/SuChecker/Python/train/'
test_path = 'C:/Users/acsel/source/repos/SuChecker/Python/validation/'

train_generator = train_datagen.flow_from_directory(train_path,
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    class_mode='categorical')
test_generator = test_datagen.flow_from_directory(test_path,
                                                  target_size=(224, 224),
                                                  batch_size=32,
                                                  class_mode='categorical')
```

Рисунок 4.2 — Фрагмент опису даних для навчання мережі

Для налаштування та навчання мережі була створена група навчальних даних та група перевірки. В ході навчання розмір даних становив 2000 зображень обличчя людини, а розмір тестової групи - 800 зображень.

Як тільки мережева архітектура побудована, вона зв'язується із групою навчальних і перевірочних даних. Було також створено зручний якір для нейронної мережі. Після налаштування середовища розпочнеться навчання на 100 епох моделі мережі з інформаційними пакетами даних розміром 224x224 точок з параметрами відображення.

Для навчання нейронної мережі було використано алгоритм стохастичного градієнтного спуску. Для оновлення ваг під час налаштування нейронної мережі було використано оптимізатор на основі обчислення градієнтів, який мінімізує функції втрат.

Вибираємо коефіцієнт навчання рівним 0.005, який визначає, наскільки кожне оновлення змінить свою вагу. Значення коефіцієнта 0,005 є досить поширеним вибором і забезпечує помірну процедуру оновлення. Коефіцієнт інерції вибрано 0.9, що допомагає оптимізатору ігнорувати шум і наближатися до мінімуму. Значення 0,9 може значно збільшити вплив попередніх градієнтів на оновлення параметрів, сприяючи більш плавному та швидкому відображенню. Коефіцієнт ослаблення шкали вибрано рівним 0.0005 і який використовується для регулювання та протидії надмірному дотриманню правил шляхом покарання за великі ваги.

Потім визначається кількість періодів навчання, які показують, скільки разів алгоритм має пройти весь набір навчальних даних. У кожній епосі модель переводиться в режим навчання, що важливо для правильного розрахунку градієнтів і подальшого оновлення ваг. Оптимізатор встановлюється на нуль перед кожним проходом, щоб уникнути накопичення градієнтів за попередні роки. Після розрахунку втрат, пов'язаних з моделлю, втрати підсумовуються, помилки поширюються в зворотному напрямку, і ваги моделі можуть бути оновлені відповідно до розрахованих градієнтів. Після завершення проходження всіх даних у навчальному наборі модель перевіряється в наборі даних перевірки.

Процес навчання нейронної мережі YOLOv8 за допомогою згенерованого набору відбувається ітераційно. На кожній епосі тренінгу можна побачити результати проміжних етапів налаштування моделі. Вони недостатньо інформативні і призначені для представлення даних про конкретний пакет вихідних даних.

Коли навчання мережі завершено, відображається загальна інформація про результати навчання, кількість пройдених епох, витрачений час навчання.

На рисунку 4.3 показані результати декількох послідовних етапів навчання.

Порівняння результатів початкового навчання із остаточним навчанням робимо висновок про те, що навчання нейронної моделі за підготовленою базою даних має відмінне підвищення ефективності моделі по відношенню до неї.

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
3/99	9.33G	0.09557	0.1112	0.1363	122	640:	100%	63/63	[01:00<00:00, 1.04it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95:	100%	32/32 [00:23<00:00, 1.39it/s]
	all	1000	6909	5.73e-05	0.000588	3.03e-05	1.04e-05		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
4/99	9.33G	0.09183	0.1072	0.1288	178	640:	100%	63/63	[00:59<00:00, 1.05it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95:	100%	32/32 [00:24<00:00, 1.29it/s]
	all	1000	6909	0.000299	0.00209	0.000179	3.71e-05		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
5/99	9.33G	0.08954	0.1086	0.1195	353	640:	100%	63/63	[00:59<00:00, 1.05it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95:	100%	32/32 [00:26<00:00, 1.21it/s]
	all	1000	6909	0.00015	0.00497	0.000111	2.62e-05		

Рисунок 4.3 — Етапи тренування та проходження навчальних епох

Аналіз отриманих даних на валідаційному наборі для оцінки завершення навчання показує, що набір даних був добре сформований, і підтверджує, що на виході з навчального процесу отримано бажаний результат. На підставі цього робимо висновок, що завдання налаштування мережі на створеній базі навчання завершено.

У магістерській роботі були виконані експериментальні дослідження по вимірюванню часу, який затрачує система спостереження для пошуку і розпізнавання особи із наявної бази. Програмним способом вимірювався інтервал часу від моменту виділення одного відеокадру і до моменту формування повідомлення про завершення процесу розпізнавання особи, яка знаходилась у зоні спостереження відеокамери.

При розпізнаванні особи були отримані значення від 75 мс до 85 мс, що для роботи системи відеоспостереження із незначним числом людей для розпізнавання є достатнім.

4.2 Аналіз якості роботи навченої моделі нейронної мережі

Для оцінки якості роботи програмного комплексу розпізнавання особи

застосовувались критерій швидкості розпізнавання по відношенню до моменту отримання відео кадру. Швидкість роботи системи розпізнавання особи демонструє затрачений час на правильне розпізнавання обличчя особи із отриманого відеопотоку. У дослідженнях вимірювався час від моменту отримання виділеного кадру із відео потоку до моменту прийняття рішення про розпізнавання особи. Перевірялися два варіанти роботи системи розпізнавання: із використанням засобів попереднього виділення зони відео зображення із знайденим обличчям людини та без використання виділення такої зони. Розпізнавання проводилось на подібних зображеннях обличчя людей, що були підготовлені у вигляді фотографій із різними кутами нахилу обличчя людини. Програмним способом фіксувався момент часу отримання відеокадру та момент часу формування повідомлення про завершення процесу розпізнавання. Вимірювання проводились для 10 різних фотографічних зображень. Дані цих вимірювань приведені у таблиці 4.1.

Таблиця 4.1 — Результати вимірювання часу на процес розпізнавання

Порядк овий номер фото	Час розпізнавання особи способом 1, мс	Час розпізнавання особи способом 2, мс
1	80	82
2	79	81
3	81	82
4	83	84
5	79	81
6	82	83
7	78	80
8	83	84
9	81	83
10	78	80
Серед. час	80,4	82,0

Розроблена програма тестувалася для різних відстаней між відеокамерою та користувачем. Перевірка роботи програми виконувалась в

умовах різного освітлення, для різних кутів нахилу та повороту обличчя людини (таблиця 4.2, рисунок 4.4).

Для тестування розробленої програми було відібрано серію фотографій зображення обличчя людей з різними кутами нахилу обличчя. Як видно на рисунку 4.3, на результат розпізнавання впливає кут огляду, тобто камера відеоспостереження фіксує положення обличчя людини у конкретний момент часу, а вже система виконує процес розпізнавання. У випадку значного кута нахилу обличчя не виділяється, що є певним недоліком запропонованого рішення.

Таблиця 4.2 — Результати розпізнавання відносно кута нахилу обличчя

Кут нахилу обличчя особи	Відсоток розпізнавання особи
5°	100%
10°	100%
15°	100%
20°	100%
25°	99%
30°	92%
35°	80%
40°	60%
45°	40%

Запропонований підхід до процесу розпізнавання особи за отриманим зображенням обличчя людини можна назвати досить точним та швидкодіючим, також із зменшенням відстані між особою та веб-камерою точність розпізнавання збільшується. Отримані результати досліджень показують, що запропонований спосіб розпізнавання особи працює досить добре та коректно. Це можна пояснити тим, що результати, отримані за допомогою запропонованого підходу, є більш точними та отримуються за менший проміжок часу, а також підтверджують, що даний підхід може бути використаний у системах відеоспостереження та розпізнавання людей у заданій зоні.

У підсумку можна зробити висновок, що запропонована послідовність

процесу розпізнавання особи за отриманим зображенням обличчя може знайти використання у системах відеоспостереження та розпізнавання людей.

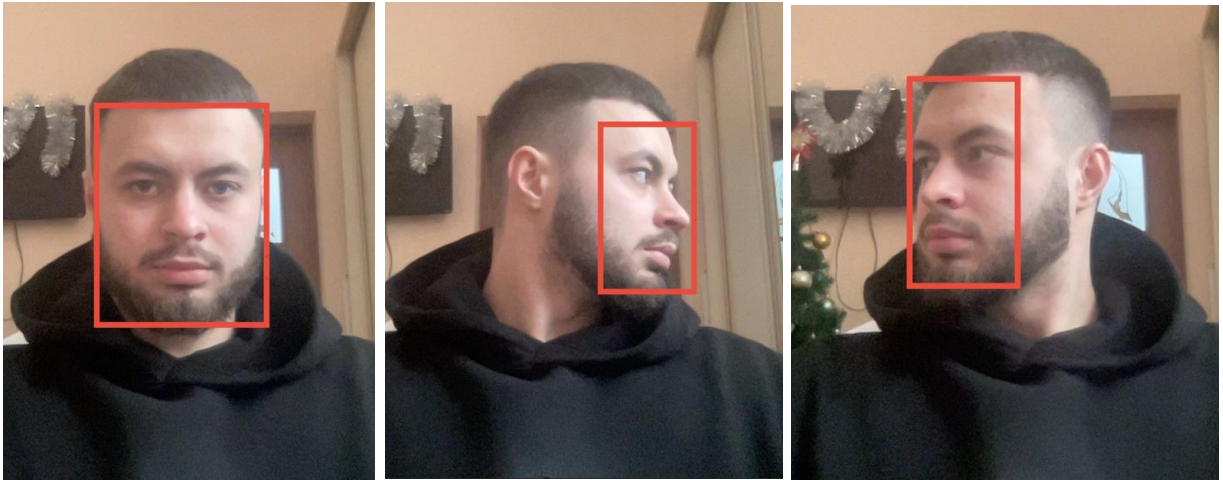


Рисунок 4.4 — Приклади тестування програми

У даному розділі магістерської роботи було проведено тестування розробленого програмного продукту та виконані експериментальні дослідження по визначенню переваг запропонованого підходу.

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ РОЗПІЗНАВАННЯ ОСОБИ У ВІДЕОЗОБРАЖЕННІ

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного комплексу для пошуку і розпізнавання особи у відеозображенні. Метою є підвищення достовірності роботи програмного комплексу для розпізнавання особи у відеозображенні. Особливістю розробки є удосконаленні методу пошуку та розпізнавання особи у відеозображенні з використанням нейронної мережі.

Аналогом може бути IP-камера IPC 2B5050s-RSM-80 (AI) ціною 7000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1 [36].

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку

Продовження табл. 5.1

Ринкові переваги					
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Закінчення табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	41	40	42
Середньоарифметична сума балів	$(41+40+42) / 3 = 41$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок удосконалення методу пошуку та розпізнавання особи у відеозображенні з використанням нейронної мережі.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів за місяць, 20 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	35000	1750,00	28	49000,000
Програміст	33000	1650,00	28	46200,000
Всього				95200,00

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 11,5 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 11,5 \% / 100 \% \quad (4.2)$$

$$З_д = (95200,00 \cdot 11,5 \% / 100 \%) = 10948,00 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (5.3)$$

$$Н_з = (95200,00 + 10948,00) \cdot 22 \% / 100 \% = 23352,56 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{вик}}{12} \text{ [грн.]}. \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ — термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 38000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,40 міс.

$$A_{обл} = \frac{38000}{2} \times \frac{1,4}{12} = 2216,67 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 11000$ грн.

5.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на

розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \tag{5.5}$$

де B — вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енера-Вінниця, $B = 10,42$ грн./кВт [37];

Π — встановлена потужність обладнання, кВт. $\Pi = 0,28$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$B_e = 0,9 \cdot 0,28 \cdot 8 \cdot 28 \cdot 10,42 = 588,18816 \text{ (грн.)}$$

Таблиця 5.5 — Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	38000	2	1,40	2216,667
Офісне обладнання (меблі)	20000	4	1,40	583,333
Приміщення	1200000	20	1,40	7000,000
Всього				9800,00

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за

статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_{\text{с}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (5.6)$$

де $H_{\text{ів}}$ — норма нарахування за статтею «Інші витрати».

$$I_{\text{в}} = 95200,00 * 55\% / 100\% = 52360 \text{ (грн.)}$$

5.2.8 До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нзв}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.7)$$

де $H_{\text{нзв}}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 95200,00 * 115\% / 100\% = 109480 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{\text{заг}} = 95200,00 + 10948,00 + 23352,56 + 9800,00 + 11000 + 588,19 + 52360 + \\ + 109480 = 312728,75 \text{ грн.}$$

5.2.10 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 312728,75 / 0,5 = 625457 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових

коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2024 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 4200 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 200 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 3000 шт., протягом другого року – на 2500 шт., протягом третього року на 2000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0 \cdot 200 + (4200 + 200) \cdot 3000) \cdot 0,8333 \cdot 0,28 \cdot (1 - 0,18) = 2410799,904 \text{ грн.}$$

$$\Delta\Pi_2 = (0 \cdot 200 + (4200 + 200) \cdot (3000 + 2500)) \cdot 0,8333 \cdot 0,28 \cdot (1 - 0,18) = 4630266,481 \text{ грн.}$$

$$\Delta\Pi_3 = (0 \cdot 200 + (4200 + 200) \cdot (3000 + 2500 + 2000)) \cdot 0,8333 \cdot 0,28 \cdot (1 - 0,18) = 6313999,747 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 13355066,13 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\begin{aligned} \Pi\Pi &= (2410799,904 / (1+0,1)^1) + (4630266,481 / (1+0,1)^2) + (6313999,747 / \\ &/ (1+0,1)^3) = 2191636,28 + 3826666,514 + 4743801,463 = 10762104,25 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ЗВ, \quad (5.11)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію.

Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2...5$, але може бути і більшим.

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 625457 = 1250914,99 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (5.12)$$

$$E_{абс} = 10762104,25 - 1250914,99 = 9511189,26 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з

так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_{ϵ} . Для цього використаємо формулу:

$$E_{\epsilon} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{а\delta c}}}{PV}} - 1, \quad (5.13)$$

де $T_{\text{ж}}$ — життєвий цикл наукової розробки, роки.

$$E_{\epsilon} = \sqrt[3]{(1 + 951189,26/1250914,99)} - 1 = 1,049$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09 \dots 0,14)$;

f — показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_{\text{в}} > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{\text{ок}} = \frac{1}{E_{\epsilon}}, \quad (5.15)$$

$$T_{ок} = 1 / 1,049 = 0,95 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,95 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 625457 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним. Період окупності складе близько 0,95 роки.

ВИСНОВКИ

Побудова систем пошуку й розпізнавання особи у заданій зоні спостереження із застосуванням комп'ютерного зору є актуальною задачею сьогодення. У даній магістерській роботі запропонована послідовність та розроблені програмні засоби для розпізнавання особи у отриманому відеозображенні.

У першому розділі магістерської роботи був здійснений аналіз сучасного стану методів та засобів пошуку й розпізнавання особи за отриманим цифровим зображенням обличчя людини, таких як геометричний метод, використання фільтрів Габора, приховані Марківські моделі, метод порівняння шаблонів, метод Віюлі-Джонса, метод гнучкого порівняння на графах та нейромережеві методи. Обрано підхід виділення зони із зображенням обличчя людини та використання нейронної мережі, що дозволять реалізувати завдання виділення та розпізнаванню особи за зображенням обличчя.

У другому розділі магістерської роботи запропонована послідовність обробки відеопотоку для пошуку і розпізнавання людей у вибраній сцені із застосуванням нейронної мережі. Для виявлення зображення обличчя людини виділяють в отриманому кадрі зону за допомогою детектора виділення обличчя, а для наступного розпізнавання особи приміняють згорткову нейронну мережу.

У третьому розділі магістерської роботи були обрані програмні засоби для формування програми по виявленню й розпізнаванню особи у виділеній зоні відеоспостереження із застосуванням мови програмування Python, із використанням засобів бібліотек OpenCV та NumPy та виконано налаштування нейронної мережі.

У четвертому розділі магістерської роботи була виконана перевірка роботи нейронної мережі і оцінка якості роботи створеного програмного продукту по розпізнаванню особи.

У п'ятому розділі роботи були виконані розрахунки по обґрунтуванню доцільності здійснення нового наукового вирішення поставленої задачі по виявленню й розпізнаванню особи у виділеній зоні відеоспостереження, виконані розрахунки необхідних економічних витрат для створення запропонованих засобів виявлення та розпізнавання особи у виділеній зоні відеоспостереження і визначено комерційний ефект від впровадження запропонованого програмного рішення в комп'ютерних системах відеоспостереження.

Створена програмний засіб може використовуватися у комп'ютерних системах для виявлення й розпізнавання особи у зоні відеоспостереження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кутковецький В. Я. Розпізнавання образів : навчальний посібник / В. Я. Кутковецький. — Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017. — 420 с.
2. Крістофор М. Б. Розпізнавання образів і машинне навчання — Київ: Кондор, 2008. — 489 с.
3. Комп'ютерний зір: [Електронний ресурс] URL:https://uk.wikipedia.org/wiki/computer_vision.
4. Царьов Р.Ю. Біометричні технології: навч. посіб. / Р.Ю. Царьов, Т. М. Лемеха. — Одеса: ОНАЗ ім. О.С. Попова, 2016. — 140 с.
5. Juwei L. Face recognition using kernel direct discriminant analysis algorithms. / L. Juwei, N. P. Konstantinos, A. Venetsanopoulos // IEEE Transactions On Neural Networks, January 2003, vol.14, no. 1, pp.117—126.
6. Bolle R.M. Guide to Biometrics /Bolle, R.M., Connell, J., Pankanti, S., Ratha, N.K., Senior, A.W.C. — New York: SUNY Press, 2004. — 241 p.
7. Nefian A.V. Face Detection and Recognition Using Hidden Markov Models / A.V. Nefian, M.H. Hayes // Image Processing. ICIP 98. International Conference, 1998, pp. 145 — 152.
8. Ahonen T. Face Description with Local Binary Patterns: Application to Face Recognition / T. Ahonen, A. Hadid, M. Pietikainen//IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, vol. 28, no 12, pp. 2037 — 2041.
9. Jones P. Rapid object detection using a boosted cascade of simple features // IEEE Conf. on Computer Vision and Pattern Recognition, 2001. V. 1, Kauai, Hawaii, USA., pp. 511—518.
10. Edwards G. J. Face recognition using active appearance models/ G. J. Edwards, T. F. Cootes, C. J. Taylor // Computer Vision . Volume 1407 of the series Lecture Notes in Computer Science, 2006. — p. 588 — 595.
11. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. — 2012. — С. 1097—1105.

12. Кардаш В. О. Програмний комплекс пошуку і розпізнавання особи у відеозображенні. / Кардаш В. О., Мартинюк Т. Б., Очуров М. А. // Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів ВНТУ. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/22377>.

13. Li, L., Mu, X., Li, S., & Peng, H. A. Review of Face Recognition Technology. IEEE Access, 2020, №8, p. 1110—1120. <https://doi.org/10.1109/access.2020.3011028>.

14. Голуб'як І. В. Методи розпізнавання облич: короткий огляд. — Україна: ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», 2020. — 78 с.

15. Т. В. Юр. Аналіз методів розпізнавання облич на зображеннях // Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка». 2015 №2(21) с. 42— 46.

16. Кривенко О. В., Трубіцина О. В. Дослідження використання методів розпізнавання обличчя людини в системах ідентифікації // Наука та виробництво. 2020. Вип 22. С. 105— 112.

17. Цивадиць Павло, Скрипник Тетяна, Вознюк Леонід. Порівняння методів виявлення об'єктів в комп'ютерному зорі. // Вісник Хмельницького національного університету, №2, 2024 (333) С. 265 — 268.

18. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. — Житомир : Вид. О. О. Євенок, 2020. — 184 с.

19. Яковенко О.О. Розробка системи розпізнавання осіб на основі згорткової нейронної мережі / О.О. Яковенко, Н.І. Кушніренко, І.С. Дорофєєва, А.Р. Євтушенко // Інформатика та математичні методи в моделюванні. 2019. Том 9, №1-2. — с. 77—87.

20. Lowe D.G. Distinctive Image Features from Scale Invariant Key points / D. G. Lowe // International Journal of Computer Vision. — 2004. —Vol. 60, № 2. — P. 91—110.

21. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. / Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. June 1 2017, № 6 (39), pp. 1137 —1149.

22. Method for identifying a person by his face (US Patent № 5164992, MPK R08Y 09/51).

23. A method for identifying a person based on the analysis of the structure of his face from the video image (US Patent № 5710833 от 08.13.2013, MPK R08Y 09/51).

24. Redmon, J. You only look once: Unified, real-time object detection / J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 779 — 788.

25. A method for traffic sign detection in an image with learning from synthetic data [Text] / A. Chigorin, G. Krivovvaz, A. Velizhev, A. Konushin // 14th International Conference Digital Signal Processing and its Applications. — 2012. — Vol. 2. — P. 316 — 335.

26. Яловега О. В., Мельник Р. А. Система виявлення обличчя на зображенні з використанням глибинної згорткової нейронної мережі. // Науковий вісник НЛТУ України. 2022, т. 32, № 2. С. 55 — 60.

27. Тимошин Ю. А. Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі / Ю. А. Тимошин, С. П. Орленко // «Адаптивні системи автоматичного управління» № 1 (32) 2018. — с. 166—173.

28. Hosang J.H. Learning non-maximum suppression. / 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21—26 July 2017 / Curran Associates, Inc, NY, 2018, pp. 6469 — 6477.

29. Wei Liu, Dragomir Anguelov, Dumitru Erhan and others. SSD: Single Shot MultiBox Detector // 2016 — 16 March [Електронний ресурс].URL: <https://storage.googleapis.com/pubtools-public-publication-data/pdf/44872.pdf>.

30. Дзюбаненко А. В. Локалізація об'єктів та вилучення основних ознак з відеопотоку в комп'ютеризованих системах телеспостереження/ А. В. Дзюбаненко // Проблеми інформатизації та управління, №4(56), 2016. — С. 34—37.

31. Колесницький О. К. Виявлення озброєних людей у відеопотоці з використанням згорткових нейронних мереж. / О. К. Колесницький, Є. В. Янковський, І. К. Денисов, І. Р. Арсенюк // Оптико-електронні інформаційно-енергетичні технології, 2023, V46, № 2, с.76 — 83.

32. Andrew G. Howard, Menglong Zhu, Bo Chen and others. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision // arXiv. — 2017, 17 April.

33. OpenCV documentation [Електронний ресурс] — Режим доступу: http://docs.opencv.org/3.0beta/doc/py_tutorials/py_video/py_meanshift/py_meanshift.html?highlight=camshift.

34. COCO. [Електронний ресурс] — Режим доступу: <https://cocodataset.org/#home>.

35. Середовище PyCharm. [Електронний ресурс] — Режим доступу: <https://uk.wikipedia.org/wiki/PyCharm>.

36. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.

37. Тарифи на електроенергію [Електронний ресурс]. Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>.

ДОДАТОК А

Міністерство освіти та науки України

Вінницький національний технічний університет

Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н. Азаров О.Д.

"29" вересня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Програмний комплекс для пошуку і розпізнавання особи у відеозображенні

08-54.МКР.007.00.000.ПЗ

Науковий керівник: д.т.н., професор

_____ Мартинюк Т. Б.

Студент групи 1КІ-23м

_____ Кардаш В. О.

Вінниця 2024

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність дослідження пов'язана із широким застосуванням засобів відеоспостереження у місцях скупчення великої кількості людей, надання доступу у банках, на кордоні, у боротьбі із злочинністю та інших сферах діяльності людей, що пов'язана із процесом розпізнавання особи за зображенням її обличчя.

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета роботи — аналіз сучасного стану і тенденцій розвитку зарубіжних і вітчизняних розробок у сфері розпізнавання обличчя людини у засобах відеоспостереження і вдосконалення технології розпізнавання обличчя людини у засобах відеоспостереження.

2.2 Призначення розробки — створення програмного продукту для виконання процесу розпізнавання особи за зображенням її обличчя.

3 Вихідні дані для виконання МКР

3.1 Технічний опис сучасних методів розпізнавання людей в відеопотоці.

3.2 Середовище розробки ПЗ Visual Studio Code.

4 Вимоги до виконання МКР

4.1 Провести аналіз різних типів технологій виявлення і розпізнавання особи у відеозображенні.

4.2 Вивчити принцип роботи та особливості систем розпізнавання особи у відеозображенні.

4.3 Запропонувати метод розпізнавання особи у відеозображенні.

4.4 Розробити технологію виявлення і розпізнавання особи у відеозображенні.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз методів та засобів виділення особи у відеозображенні	20.09.2024р.	30.09.2024р.	Розділ 1
2	Розробка нейромережевої технології виявлення і розпізнавання особи у відеозображенні	01.10.2024р.	04.10.2024р.	Розділ 2
3	Розробка програми виявлення і розпізнавання особи у відеозображенні	05.10.2024р.	18.10.2024р.	Розділ 3
4	Перевірка працездатності програмного продукту	19.10.2024р.	01.11.2024р.	Розділ 4
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання особи у відеозображенні	02.11.2024р.	15.11.2024р.	Розділ 5
6	Апробація та впровадження результатів дослідження	16.11.2024р.	30.11.2024р.	Тези доповіді
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	02.12.2024р.	10.12.2024р.	ПЗ, графічний матеріал
8	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	10.12.2024р.	12.12.2024р.	Оформлені документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами.

Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02- П.001.01:21».

ДОДАТОК Б

Методи розпізнавання обличчя людини

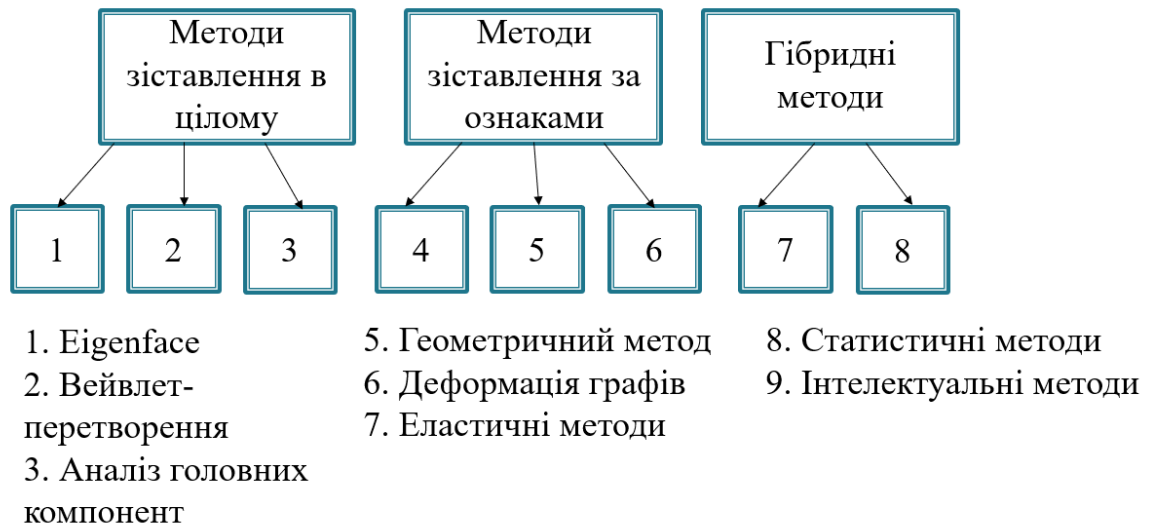


Рисунок Б.1 — Методи розпізнавання обличчя людини

ДОДАТОК В

Типи нейронних мереж

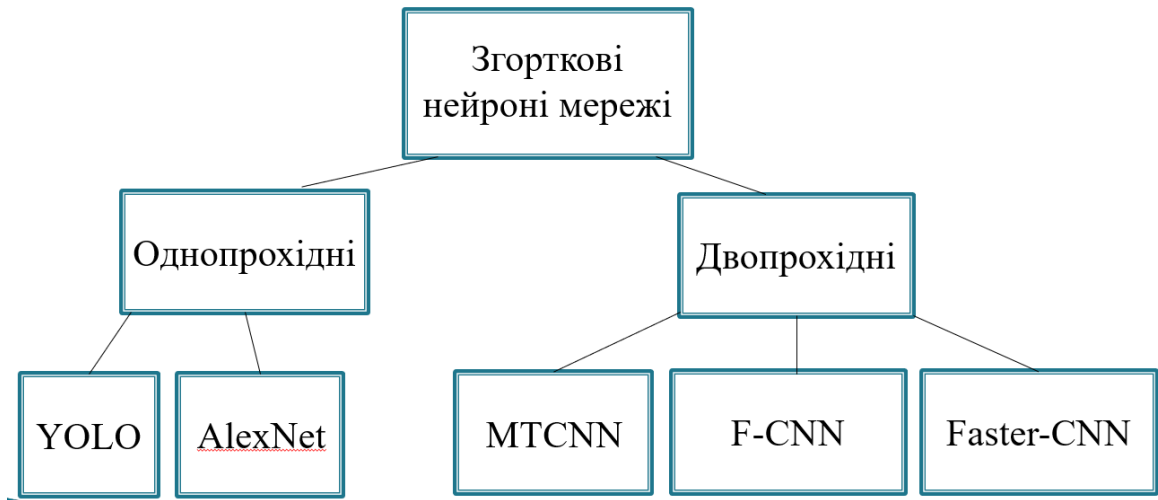
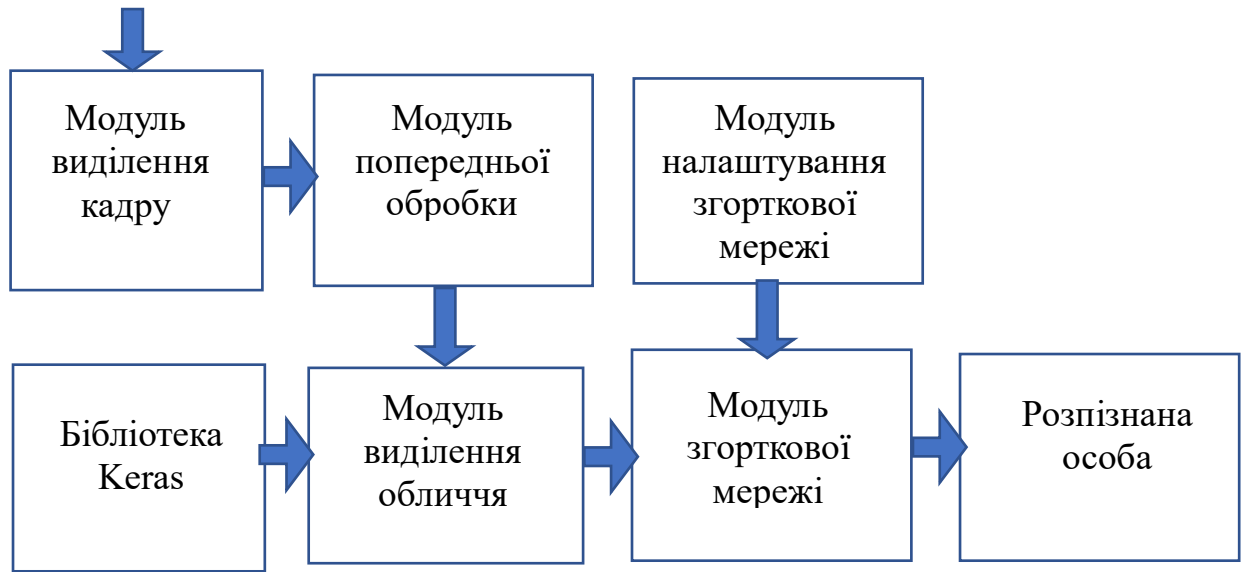


Рисунок В.1 — Типи нейронних мереж

ДОДАТОК Г**Структура програми розпізнавання особи****Рисунок Г.1 — Структура програми розпізнавання особи**

ДОДАТОК Д

Лістинг програми виділення кадрів

```

from imutils.video import VideoStream
import dlib
import cv2
import face_recognition
import numpy as np

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(
    shape_predictor_face_landmarks.dat)

def __initialize_camera(self):
    self.video_capture = cv.VideoCapture(self.camera_index)
    if not self.video_capture.isOpened():
        return False
    self.video_capture.set(cv.CAP_PROP_FRAME_WIDTH,
        self.camera_resolution[0])
    self.video_capture.set(cv.CAP_PROP_FRAME_HEIGHT,
        self.camera_resolution[1])
    return True
def run(self):
    self.is_running = True
    if not self.__initialize_camera():
        self.camera_initialized.emit(False)
        self.is_running = False
        self.video_capture = None
        return
    self.camera_initialized.emit(True)
    while self.is_running:
        is_successful_camera_frame_read, camera_frame =
            self.video_capture.read()
        if is_successful_camera_frame_read:
            self.camera_frame_read.emit(camera_frame)
            if self.is_person_location_detection_running:
                self.camera_frames_to_process.put(camera_frame)
            self.camera_frames_to_process.join()
        self.video_capture.release()

def __initialize_detection_model(self):
    self.detection_model =

```

```

cv.dnn_DetectionModel(self.detection_model_weights_file_path,
self.detection_model_configuration_file_path)
self.detection_model.setPreferableBackend(cv.dnn.DNN_BACKEND_CUDA)
self.detection_model.setPreferableTarget(cv.dnn.DNN_TARGET_CUDA)
self.detection_model.setInputParams(self.detection_model_input_scale, self.detection_model_input_size)

def __detect_camera_frame_objects_and_measure_fps(self,
camera_frame_to_process):
    start_detection_time = time.time()
    class_ids, confidences, bounding_boxes =
self.detection_model.detect(
    camera_frame_to_process,
self.detection_model_confidence_threshold,
self.detection_model_nms_threshold)
    end_detection_time = time.time()
    fps_number = 1 / (end_detection_time -
start_detection_time)
    return class_ids, confidences, bounding_boxes,
fps_number

```

ДОДАТОК Е

Лістинг програми налаштування нейронної мережі

```

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np from sklearn.metrics
import accuracy_score
import numpy as np

def cost_function(predicted, actual):
    return np.mean(np.abs(predicted - actual))
def G_function(previous_function, predicted_features, kernel):
    return previous_function / (predicted_features @ kernel)
def g_function(J, kernel, G, features):
    return J / (np.sum(G) * features @ kernel)
def h_function(J, features, G, kernel):
    return J / (np.sum(G) * features @ kernel)
def update_weights(previous_weights, g_function_result):
    return previous_weights - g_function_result
def update_features(previous_features, h_function_result):
    return previous_features - h_function_result
def plot_loss_accuracy_combined(cnn):
    epochs = range(1, len(cnn.train_accuracies) + 1)
    train_losses = cnn.train_losses
    train_accuracies = cnn.train_accuracies
    test_losses = cnn.test_losses
    test_accuracies = cnn.test_accuracies
    plt.figure(figsize=(8, 6))
    plt.scatter(train_accuracies, train_losses, color='blue', label='Training',
marker='o')
    plt.scatter(test_accuracies, test_losses, color='green', label='Testing',
marker='o')
    plt.title('Loss-Accuracy Relationship')
    plt.xlabel('Accuracy')
    plt.ylabel('Loss') plt.legend()
    plt.show()
def plot_loss_accuracy_fixed(cnn):
    # Отримання середніх значень точності та втрат
    average_train_losses = np.mean(cnn.train_losses)
    average_train_accuracies = np.mean(cnn.train_accuracies)
    average_test_losses = np.mean(cnn.test_losses)
    average_test_accuracies = np.mean(cnn.test_accuracies)
    plt.figure(figsize=(12, 5))

```

```

plt.subplot(1, 2, 1)
plt.plot(range(1, len(cnn.train_losses) + 1), cnn.train_losses, label='Loss',
marker='o')
plt.axhline(y=average_train_losses, color='r', linestyle='--', label='Average
Train Loss')
plt.title("Training Loss")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.subplot(1, 2, 2)
plt.plot(range(1, len(cnn.train_accuracies) + 1), cnn.train_accuracies,
label='Accuracy', marker='o', color='orange')
plt.axhline(y=average_train_accuracies, color='r', linestyle='-- ',
label='Average Train Accuracy')
plt.title("Training Accuracy") plt.xlabel('Epochs')
plt.ylabel('Accuracy') plt.legend() plt.tight_layout()
plt.show()
print(f"Average Train Loss: {average_train_losses}")
print(f"Average Train Accuracy: {average_train_accuracies}")
print(f"Average Test Loss: {average_test_losses}")
print(f"Average Test Accuracy: {average_test_accuracies}")

class MyFirstCNN:
def __init__(self, input_size, hidden_layers, output_size, kernel_size=3):
self.input_size = input_size
self.hidden_layers = hidden_layers
self.output_size = output_size
self.kernel_size = kernel_size
self.train_accuracies = [] self.train_losses = []
self.test_accuracies = []
self.test_losses = []
self.layers = []
input_layer = {"weights": np.random.randn(input_size, hidden_layers[0]),
"features": np.zeros((1, hidden_layers[0]))}
self.layers.append(input_layer)
for i in range(len(hidden_layers) - 1): hidden_layer = {"weights":
np.random.randn(hidden_layers[i], hidden_layers[i + 1]), "
features": np.zeros((1, hidden_layers[i + 1]))}
self.layers.append(hidden_layer)
output_layer = {"weights": np.random.randn(hidden_layers[-1],
output_size),
"labels": np.zeros((1, output_size)),
"values": np.zeros((1, output_size))}
self.layers.append(output_layer)

```

```

def add_layers(self, num_layers, layer_size):
    for _ in range(num_layers):
        new_layer = {"weights": np.random.randn(self.hidden_layers[- 1],
layer_size),
        "features": np.zeros((1, layer_size))}
        self.layers.insert(-1, new_layer)
        def set_output_layer_params(self, labels_size, values_size):
            self.layers[-1]["weights"] = np.random.randn(self.hidden_layers[- 1],
labels_size)
            self.layers[-1]["labels"] = np.zeros((1, labels_size))
            self.layers[-1]["values"] = np.zeros((1, values_size))
            def set_hidden_layer_params(self, layer_index, kernel_size,
hidden_features_size):
                self.layers[layer_index]["weights"] =
np.random.randn(hidden_features_size, kernel_size)
                self.layers[layer_index]["features"] = np.zeros((1, hidden_features_size))
            def set_output_layer_params(self, labels_size, values_size):
                self.layers[-1]["weights"] = np.random.randn(self.hidden_layers[- 1],
labels_size)
                self.layers[-1]["features"] = np.zeros((1, values_size))
        def train(self, X_train, y_train, max_difference=10.0):
            prev_loss = float('inf')
            last_losses = []
            epoch = 0
            # Зберігаємо значення точності та втрат для кожного зображення
            accuracies_per_image = []
            losses_per_image = []
            accuracy = np.mean(predicted_labels > 0.9)
            print(f"Epoch {epoch + 1}, Loss: {loss}, Accuracy: {accuracy}")
            self.train_accuracies.append(accuracy)
            self.train_losses.append(loss) while True:
                for i in range(len(X_train)):
                    x_sample, y_sample = X_train[i], y_train[i]
                    conv_result = self.convolution(self.layers[0]["weights"], x_sample)
                    predicted_labels = sigmoid(conv_result)
                    loss = cost_function(predicted_labels, y_sample)
                    # Збереження значення точності та втрати для кожного зображення
                    accuracy = np.mean(predicted_labels > 0.9)
                    accuracies_per_image.append(accuracy)
                    losses_per_image.append(loss)
                    if len(last_losses) == 3 and max(last_losses) - min(last_losses) <=
max_difference:
                        print(f"Training stopped. Last three losses within {max_difference}.")
                        break

```

```

epoch += 1
self.evaluate_on_test_set(X_test, y_test)
def predict(self, X):
    conv_result = self.convolution(self.layers[0]["weights"], X)
    detection_result = sigmoid(conv_result)
    return detection_result
pass
def detect_objects(self, testing_image, threshold=0.5):
    conv_result = self.convolution(self.layers[0]["weights"], testing_image)
    detection_result = conv_result # Без використання сигмоїди
    bounding_boxes = detection_result[0] # Припускаємо, що маємо лише
один приклад
    plt.imshow(testing_image.reshape((28, 28)), cmap='gray')
    for bbox in bounding_boxes:
        x_min, x_max, y_min, y_max = bbox
        accuracy = x_max
        # Перевірка точності порівняно з порогом
        if accuracy >= threshold:
            rect = plt.Rectangle((x_min, y_min), (x_max - x_min), (y_max - y_min),
                                edgecolor='red', facecolor='none')
            plt.gca().add_patch(rect)
            plt.show()
    return detection_result
def plot_loss_accuracy(self, losses, accuracies):
    # Відображення графіка втрат та точності відносно кількості епох
    epochs = len(losses) plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(range(1, epochs + 1), losses, label='Loss', marker='o')
    plt.title("Training Loss")
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()
    plt.subplot(1, 2, 2)
    plt.plot(range(1, epochs + 1), accuracies, label='Accuracy', marker='o',
color='orange')
    plt.title("Training Accuracy")
    plt.xlabel('Epochs') plt.ylabel('Accuracy')
    plt.legend()
    plt.tight_layout()
    plt.show()
def convolution(self, kernel, features, stride=1):
    kernel_height, kernel_width = kernel.shape
    input_height, input_width = features.shape
    height = (input_height - kernel_height) // stride + 1

```

```

output_width = (input_width - kernel_width) // stride + 1
conv_result = np.zeros((output_height, output_width))
    # Процес конволюції
    for i in range(0, input_height - kernel_height + 1, stride):
        for j in range(0, input_width - kernel_width + 1, stride):
            conv_result[i // stride, j // stride] = np.sum(features[i:i + kernel_height, j:j +
kernel_width] * kernel)
    return conv_result
    return conv_result
    def save_epoch_results(self, epoch, average_loss, average_accuracy):
        result_string = f"Epoch {epoch + 1}, Average Loss: {average_loss},
Average Accuracy: {average_accuracy}"
        print(result_string)
    def evaluate_on_test_set(self, X_test, y_test):
        accuracies_per_image = []
        losses_per_image = []
        for i in range(len(X_test)):
            x_sample, y_sample = X_test[i], y_test[i]
            conv_result = self.convolution(self.layers[0]["weights"], x_sample)
            predicted_labels = sigmoid(conv_result)
            # Обчислення функції втрат
            loss = cost_function(predicted_labels, y_sample)
            accuracy = np.mean(predicted_labels > 0.9)
            accuracies_per_image.append(accuracy)
            losses_per_image.append(loss)
            self.test_accuracies.append(accuracy)
            self.test_losses.append(loss)
            # Обчислення середніх значень для тестової вибірки
            average_loss = np.mean(losses_per_image)
            average_accuracy = np.mean(accuracies_per_image)
            # Збереження результатів тестування
            self.save_test_results(average_loss, average_accuracy)

```

ДОДАТОК Ж**Інтелектуальні методи розпізнавання**

Рисунок Ж.1 — Інтелектуальні методи розпізнавання

ДОДАТОК И

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: _____ Програмний комплекс для пошуку і розпізнавання особи у відеозображенні _____

Тип роботи: _____ Магістерська кваліфікаційна робота _____
(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ _____ кафедра обчислювальної техніки _____
(кафедра, факультет (інститут), навчальна група)

Керівник _____ Мартинюк Т.Б., проф. каф. ОТ _____
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	91%
Загальна схожість	9%

Аналіз звіту подібності (відмітити потрібне)

- ✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Кардаш В.О. _____
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

Особа, відповідальна за перевірку _____ Захарченко С.М. _____
(підпис) (прізвище, ініціали)

Експерт _____
(за потреби) (підпис) (прізвище, ініціали, посада)