

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ
ЛЮДЕЙ У ВІДЕОПОТОЦІ**

Виконав: студент 2 курсу, групи 2КІ-22м
спеціальності 123 — «Комп'ютерна інженерія»

В Дигодій В. І.

Керівник: д.т.н., проф. каф. ОТ

Т. Мартинюк Мартинюк Т. Б.

« 14 » 12 2023 р.

Опонент: к.т.н., доц. каф. ПЗ

Г. Черноволик Черноволик Г. О.

« 18 » 12 2023 р.

Допущено до захисту

Завідувач кафедри ОТ

О.Д. Азаров д.т.н., проф. Азаров О.Д.

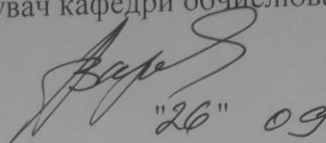
" 20 " 12 2023 р.

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
2023 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Дигодію Віктору Івановичу

1 Тема роботи «Нейромережева технологія виявлення і розпізнавання людей у відеопотоці» керівник роботи Мартинюк Тетяна Борисівна д.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу від «18» вересня 2023 року № 247.

2 Строк подання студентом роботи: 08.12.23.

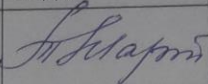
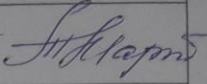
3 Вихідні дані до роботи: технічний опис сучасних методів та засобів розпізнавання людей в відео потоці, середовище розробки ПЗ Visual Studio Code.

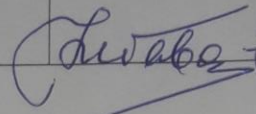
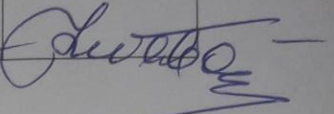
4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз методів та засобів виділення людей у відеопотоці, розробка нейромережевої технології виявлення і розпізнавання людей у відеопотоці, розробка програми виявлення і розпізнавання людей у відеопотоці, перевірка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

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): блок-схема алгоритму інформації про обличчя на форму, блок-схема алгоритму формування списків розпізнаних та нерозпізнаних обличь, UML-діаграма класів програмного засобу.

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	02.11.23	02.12.23


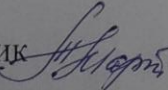
 

7 Дата видачі завдання 20.09.23

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	03.10.2023р.	виконано
2	Огляд та аналіз існуючих підходів щодо побудови нейромережевої технології виявлення і розпізнавання людей у відеопотоці	14.10.2023р.	виконано
3	Результати проведених досліджень та методів виявлення і розпізнавання людей у відеопотоці	25.10.2023р.	виконано
4	Проектування нейромережевої технології виявлення і розпізнавання людей у відеопотоці	19.11.2023р.	виконано
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання людей у відеопотоці	02.12.2023р.	виконано
6	Оформлення пояснювальної записки і презентації	04.12.2023р.	виконано
7	Попередній захист	07.11.2023р.	виконано
8	Перевірка «антиплагіат»	05.12.2023р.	виконано
9	Аналіз виконання роботи, висновки, додатки	05.12.2023р.	виконано
10	Оформлення пояснювальної записки та ілюстративного матеріалу	06.12.2023р.	виконано
11	Перевірка якості виконання магістерської роботи та усунення недоліків	07.12.2023р.	виконано

Студент  Дигодій Віктор ІвановичКерівник  д.т.н., професор Мартинюк Тетяна Борисівна

АНОТАЦІЯ

УДК 004.93

Дигодій В. І. Нейромережева технологія виявлення і розпізнавання людей у відеопотоці. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма Комп'ютерна інженерія. Вінниця: ВНТУ, 2023. — 108 с.

На укр. мові. Бібліогр.: 34 назв; рис. 26; табл. 8.

Магістерську кваліфікаційну роботу присвячено розробці нейромережевої технології виявлення і розпізнавання людей у відеопотоці. Здійснено огляд існуючих методів та засобів виявлення та розпізнавання людей у відеопотоці, визначено їх переваги та недоліки. Розроблено нейромережеву технологію виявлення і розпізнавання людей у відеопотоці, в основі якої знаходиться використання моделі активної форми у вигляді ключових точок та форми контуру обличчя людини для її розпізнавання. Розроблено програму виявлення і розпізнавання людей у відеопотоці 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, контур об'єкта, згорткова нейронна мережа.

ANNOTATION

Dygodij V. I. Master's thesis in specialty 123 - computer engineering, computer engineering education program. Vinnytsia: VNTU, 2023. — 109 p.

In Ukrainian language. Bibliographer: 34 titles; fig.: 26; table 8.

The master's thesis is devoted to neural network technology for detecting and recognizing people in a video stream. A review of existing methods and tools for detecting and recognizing people in a video stream was conducted, their advantages and disadvantages were identified. A study was conducted that allowed us to analyze existing ones and propose our own neural network technology for detecting and recognizing people in a video stream. Programs for detecting and recognizing people in a video stream have been developed. An economic calculation of the feasibility of implementing the development was carried out. A neural network was trained to recognize a given number of persons. The created software application was tested to determine its performance. An economic calculation of the feasibility of implementing a new development was carried out, the financial costs were calculated and the economic benefits from the implementation of the proposed solution in the form of a completed software product were determined.

Key words: video stream, object contour, convolutional neural network.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ	12
1.1 Поняття штучного інтелекту для відеоспостереження	12
1.2 Комп'ютерний зір як засіб виділення людей	15
1.3 Аналіз способів пошуку та виділення обличчя	19
1.4 Нейромережа глибокого навчання	28
2 РОЗРОБКА НЕЙРОМЕРЕЖЕВОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ	35
2.1 Етапи формування пошуку обличчя в відео потоці	35
2.2 Підготовка та обробка даних по розпізнаванню обличчя	40
2.3 Розробка архітектури нейромережі	42
3 РОЗРОБКА ПРОГРАМИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ	50
3.1 Розробка структури програми розпізнавання людей	50
3.2 Розробка програми розпізнавання людей	52
3.3 Навчання нейронної мережі	56
4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ..	62
4.1 Перевірка якості роботи програми	62
4.2 Оцінка ефективності роботи програми	65

					<i>08-54.МКР.005.00.000 ПЗ</i>			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Дигодій В. І.</i>			Нейромережева технологія виявлення і розпізнавання людей у відеопотоці	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Мартинюк Т. Б.</i>					6	109
<i>Опонент</i>		<i>Черноволик Г. О.</i>				ВНТУ, гр. 1КІ-22м		
<i>Н. Контроль</i>		<i>Швець С.І.</i>						
<i>Затверджую</i>		<i>Азаров О.Д.</i>						

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ	72
5.1 Проведення комерційного та технологічного аудиту	72
5.2 Розрахунок кошторису витрат на розробку програм	75
5.3 Розрахунок експлуатаційних витрат для нової розробки	81
5.4 Розрахунок економічного ефекту від розробки програми	83
ВИСНОВКИ.....	88
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	90
ДОДАТОК А Технічне завдання.....	94
ДОДАТОК Б Блок-схема алгоритму інформації про обличчя на форму.....	98
ДОДАТОК В Блок-схема алгоритму формування списків розпізнаних та нерозпізнаних обличч.....	99
ДОДАТОК Г UML-діаграма класів програмного засобу.....	100
ДОДАТОК Д Лістинг програми з тренування нейромережі.....	101
ДОДАТОК Е Фрагмент програмного коду	105
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	109

ВСТУП

Розпізнавання людей — це актуальна й важлива область дослідження у сучасному світі, і вона впливає на різні сфери нашого життя. Ось кілька основних аспектів, що підкреслюють актуальність цього дослідження. Розпізнавання обличь і біометричні системи стають все більш важливими для забезпечення безпеки та контролю доступу [1]. Вони використовуються в аеропортах, на кордонах, в банках, наряду з іншими місцями, де важливо визначити особу. В сучасному світі існує зростаюча загроза ідентифікаційного шахрайства та підробки. Розпізнавання обличь може використовуватися для забезпечення більшої кібербезпеки, такою як розпізнавання особи для автентифікації в онлайн-банкінгу. Розпізнавання обличь грає важливу роль в розвитку автономних автомобілів, оскільки вони повинні розпізнавати пасажирів та інших учасників дорожнього руху [2]. У сфері медицини розпізнавання обличь може бути використане для розпізнавання пацієнтів, слідкування за станом пацієнтів, а також для виявлення захворювань та медичних аномалій. Розпізнавання обличь також має важливе значення у сфері соціальних мереж та маркетингу. Вони можуть використовуватися для аналізу поведінки користувачів та персоналізації реклами. Розпізнавання обличь може бути важливим інструментом у боротьбі зі злочинністю, таким як розшук злочинців та попередження тероризму. У галузі медіа та розваг розпізнавання обличь використовується для розпізнавання акторів на екрані, створення анімаційних персонажів та виробництва відеоігор [3]. Вивчення розпізнавання обличь є важливою галуззю досліджень і навчання в області комп'ютерного зору та штучного інтелекту.

Зростаючі обчислювальні можливості та розвиток методів машинного навчання, зокрема нейромереж, дозволяють значно покращити точність та ефективність систем розпізнавання обличь [4]. Отже, дослідження в цій області має великий потенціал і важливе значення для нашого сучасного суспільства.

На цей час різноманітні цифрові технології обробки даних є неймовірно поширеними через розвиток комп'ютерної технології. Зокрема, особливо сучасними є системи комп'ютерного зору й аналізу відео потоків. Дані системи

можуть застосовуватися при обороні від несанкціонованого доступу на особистими території та організації, пошуку свобідного місця для транспорту при паркуванні, розпізнаванні обличчя персони, наприклад у відео домофонах, визначенні номерів автомобіля під час порушення дорожнього руху. Таким чином, при введеній в дію систем комп'ютерного зору головним завданням є розпізнавання об'єктів. У тому випадку, якщо для людини розпізнавання фігур є простим завданням, тоді для комп'ютерної системи все не так просто. Одним серед важливих завдань є виявлення оптимальних зображень відносно критеріїв якості виявлення і швидкістю маніпулювання [5]. Також при опрацюванні відео послідовностей необхідно знайти цифрову відповідність серед двома сусідніми зображеннями.

Ключовим є застосування інтелектуального відеоспостереження, яке застосовує штучний інтелект і нейронні мережі для розв'язання багатьох задач розпізнавання візуальних відображень. Для інтелектуальних систем відеоспостереження може використовуватися резервна камера слідкування, на яку можна налаштуватися при втраті об'єкта із області спостереження. Ще одним із способів використання систем відеоспостереження є обробка відео потоків при нагляді дорожньо-транспортного трафіку, безпеки на власних територіях та в відділах з обмеженим доступом, використанні автоматизованого керуванні на промислових підприємствах [6, 7].

Оскільки системи аналізу відео потоку є надзвичайно актуальними на цей час, то в даній роботі будуть розглянуті методи та алгоритми розпізнавання об'єктів, що застосовує штучний інтелект, нейронні мережі і машинне навчання, крім того проведено аналіз систем відеоспостереження та розпізнавання предметів в реальному часі.

Отже, виходячи із викладеного, задачі подальшого розроблення й вдосконалення методів пошуку, виявлення й розпізнавання людей у відео потоці, що є кращими за такими критеріями як достовірність та обсяг задіяних ресурсів, на теперішній час є **актуальними**.

Метою дослідження є вдосконалення технології розпізнавання обличчя

людини у засобах відеоспостереження.

Задачі дослідження. Виконати аналіз існуючих методів та ліквідувати неточності обробки відео зображень для пошуку та розпізнавання обличь. Запропонувати поліпшену технологію обробки відео зображень для пошуку та розпізнавання обличчя. Розробити алгоритм та створити програму обробки отриманого відео зображення з метою виділення та розпізнавання обличчя людини. Виконати обґрунтування доцільності впровадження нового наукового рішення, реалізувати розрахунок економічних затрат для реалізації програмних засобів виокремлення обличь з цифрових зображень та проаналізувати економічні вигоди від впровадження нового створеного програмного рішення.

Об'єкт дослідження в області розпізнавання обличь — це процес виявлення та розпізнавання обличчя людини у отриманому відео потоці. Ця область належить до комп'ютерного зору і обробки зображень та має великий спектр застосувань.

Предмет дослідження розпізнавання обличь — це методи обробки відео зображень для пошуку, виділення й розпізнавання особи за отриманим та виділеним відео зображенням обличчя людини, що дозволяють комп'ютерам автоматично аналізувати й ідентифікувати обличчя людей у цифрових зображеннях та відео.

Методи дослідження магістерської роботи: метод аналітичного огляду, метод порівняння і аналогій, метод моделювання, метод об'єктно-орієнтованого програмування для розробки програмного додатку. Використовувались методи теорії множин для формування множини ознак для розпізнавання, методи диференційного числення для налаштування мережі, методи математичної статистики для виконання аналізу отриманих результатів розпізнавання. У роботі використано принципи об'єктно-орієнтованого програмування для реалізації запропонованого підходу.

Новизна одержаних результатів полягає у тому, що удосконалено технологію обробки відео зображень для виділення й розпізнавання особи, яка відрізняється від існуючих виконанням пошуку й виділення обличчя людини у

вибраній сцені відео зображення із використанням характерних рис обличчя у вигляді контуру обличчя й ключових точок обличчя, що дозволяє із більшою точністю проводити виділення й розпізнавання особи в отриманому відео потоці.

Практичне значення одержаних результатів:

— розроблено алгоритм обробки відео зображень для виділення рис обличчя людини шляхом знаходження контуру;

— розроблено програму обробки відео зображень для розпізнавання людей із урахуванням характерних особливостей рис обличчя людини та знаходження ключових точок обличчя людини.

Апробація результатів магістерської роботи: зроблено доповідь на Всеукраїнській ЛШ науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ—2024).

За результати магістерської роботи **опубліковано** 1 наукову працю [8]:

Дигодій В. І. Нейромережева технологія виявлення і розпізнавання людей у відеопотоці. / В. І. Дигодій, Т. Б. Мартинюк, М. А. Очуров // Матеріали ЛШ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ-2024)». — Електрон. текст. дані. — Вінниця, 2024. — Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2024/paper/view/19561/16206>.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ

Цей розділ магістерської роботи присвячений огляду й аналізу існуючих методів виділення людей у зоні відеоспостереження та їх розпізнавання.

1.1 Поняття штучного інтелекту для відеоспостереження

Завдання розпізнання обличчя на зображенні нерідко є першим кроком у етапі розв'язання задачі вищого рівня важності — розпізнавання персони, деталей особи або її виразу обличчя. Крім того, данні про присутність та чисельність осіб на зображенні можливо бути практичною в системах автоматичного реєстрування кількості відвідувачів; системах пропускового нагляду в установах, аеропортах і метро; автоматичних системах передбачення нещасних випадків; інтелектуальних інтерфейсах «людина-комп'ютер»; в фототехніці заради автоматичного фокусування на обличчі персони, а також заради стабілізації зображення персони з метою спрощення розпізнавання емоцій; з метою розширення області стерео бачення під час розробки систем 3D відображення. Те, що для людей є фотографією, для системи ідентифікації образів — лише комплект пікселів з різноманітними параметрами кольору. Щоб тренувати систему розпізнавати на зображенні певні об'єкти, необхідно дати їй датасета — сукупність з тисяч зображень, де зазначено, в яких саме знаходиться необхідний об'єкт. Наприклад, коли ми хочемо, щоб система навчилася знаходити на знімках людей, необхідно показати їй велика кількість фотографій людей різноманітного віку, в різних позах та одязі, в відмінних умовах [8]. Після даного тренування система буде здатна безпомилково розпізнати персон на фотографіях.

В останні роки і виробники продукції для систем відеоспостереження, і інсталятори наперебій говорять про штучний інтелект (ШІ/AI), Технології Глибинного Машинного Навчання (ТГМН) та про використання їх у системах електронної безпеки. І справді — ці технології активно використовуються в багатьох функціях інтелектуальної відеоаналітики. Штучний інтелект допомагає

розпізнавати обличчя та автомобільні номери, допомагає вести стеження з камер відеоконтролю за переміщенням окремо взятої людини, і ще багато чого. Тому і ставлення до використання штучного інтелекту в системах відеоспостереження в різних людей неоднозначне — одні «за», інші — різко «проти».

Сучасний штучний інтелект розподіляють на три умовні типи: системи, здатні самостійно приймати рішення; системи, які забезпечують підтримку прийняття рішень; і зрештою системи, здатні блискавично обробляти гігантські об'єми даних та вишукувати в них закономірності та відхилення від норми (аномалії).

От власне цей третій тип штучного інтелекту і використовується зазвичай в системах електронної безпеки. І проти нього нема нарікань. Як правило люди побоюються і заперечують ті типи систем штучного інтелекту, які здатні самостійно, без втручання людини, приймати важливі рішення. Це стосується насамперед робототехніки.

А ми повернемося до того типу штучного інтелекту, який здатен швидко обробляти, класифікувати та відтворювати великі об'єми відеоданих із камер спостереження. Власне на цих уміннях штучного інтелекту і побудована більша частина функцій інтелектуальної відеоаналітики в системах спостереження. Йдеться про такі технології.

Машинне навчання (наприклад, функція розпізнавання обличь IP-камерою відеоспостереження — чим більше обличь потраплятиме в базу даних гаджету, тим більше людей програма зможе ідентифікувати і знайти, якщо отримає команду на пошук).

Комп'ютерний зір — виявлення, класифікація та стеження за визначеним типом об'єктів — наприклад, камеру спостереження на автомагістралі можна віддалено запрограмувати на виявлення автомобілів червоного кольору, якщо злочинець рухається на такому авто [9]. Тож якщо камера виявить автомобіль червоного кольору, служба охорони об'єкту одразу отримає тривожне повідомлення.

Глибокий аналіз даних — структурування, аналіз та вичленення корисної

інформації із невпорядкованого масиву різномірних потоків інформації.

І, як на мене, найцікавіша технологія штучного інтелекту, що вже активно застосовується в відеоспостереженні — це задіяння так званої «нечіткої логіки», або ще кажуть «розмиті логіки». Це коли людина не дає машині чіткий алгоритм пошуку чогось, а лише окреслює контекст — сподіваючись, що машина навчиться приблизно розуміти поведінку людини, яка не завжди є адекватною і не завжди є логічною [10]. Наприклад, якщо людина довго дивиться на камеру спостереження, штучний інтелект може сприйняти це за погрозу, і відправити тривожний сигнал. Він це розуміє так — що або ця людина вандал, і хоче просто знищити камеру, або вона потенційний злодій, який роздивляється як цю камеру можна нейтралізувати.

Переваги система відеоспостереження зі штучним інтелектом.

Підстрахування або і часткова заміна людей зі служби охорони — в часи аналогового відеоспостереження камера передавала зображення на монітор, а за ними постійно слідкувала жива людина. Її завдання — вчасно побачити неадекватну чи злочинну поведінку в кадрі і швидко подати сигнал колегам, які ліквідують потенційну небезпеку. Тепер сигнали тривоги, причому найрізноманітніші, подає сам штучний інтелект автоматично, без втручання людини. Причому штучний інтелект забезпечує такі складні функції, як ідентифікація забутого/залишеного предмету, наприклад сумки, пересування в робочій зоні камери розшукуваної людини (трекінг) і тому подібне. Крім того, на відміну від живої людини, ШІ ніколи не втратить пильність і не прогавить ту задачу, яку йому призначили.

Багатозадачність — тобто відеокамери, окрім своєї прямої функції – вести відеоспостереження, виконують ще багато різних завдань, пов'язаних із відеоаналітикою: від збирання статистики (скільки людей пройшло через прохідну за день, скільки автомобілів заїхало на заправку тощо) до пропуску чи навпаки заборони пропуску окремих людей чи транспортних засобів на охоронювану територію. Деякі моделі відеокамер зі штучним інтелектом уміють навіть температуру тіла людини виміряти на віддалі.

Найбільш поширена і востребувана в відеонагляді функція штучного інтелекту — це робота із архівом. Можна вважати що штучний інтелект просто спеціалізується на роботі із великими базами даних — він прекрасно їх класифікує, впорядковує і називає окремі фрагменти. Тобто раніше, щоб знайти якийсь конкретний шмат записаного відео, потрібно було сидіти і перемотувати відеокасети [11]. Це займало час. Тепер же можна завдати пошук за певними параметрами, і штучний інтелект моментально знайде необхідне. Наприклад, «забиваємо» у пошук фото людини, і штучний інтелект перевірить весь відео архів і знайде всі відео фрагменти, в яких з'являлася розшукувана людина.

Стратегічне планування — найновіша із функцій штучного інтелекту, яка застосовується у відеоспостереженні. Візьмемо магазин чи цілий торгово-розважальний центр. Якщо грамотно розташувати відеокамери із штучним інтелектом, то можна зібрати наступну статистику: в які конкретно години в магазині найбільше людей, а в які найменше; біля яких стелажів із товаром крутиться багато людей, а до яких стелажів покупці не підходять. Більше того, так звана теплова карта покаже навіть шляхи руху потенційних покупців по всій території магазину; якого віку та статі переважна більшість людей, що заходять у магазин; скільки часу затримується в магазині середньостатистичний відвідувач і тому подібне.

Звісно ж, що всі ці статистичні дані дозволять власникам магазину розробити найбільш прийнятну маркетингову стратегію цього закладу. Більше того, сам штучний інтелект може розробити цю стратегію і викласти власникам закладу свої поради.

1.2 Комп'ютерний зір як засіб виділення людей

Комп'ютерний зір — це міждисциплінарна дослідницька область, яка досліджує якими методами можна створити комп'ютери з метою отримання високого ступення розуміння цифрових зображень чи відео. З точки зору технічної галузі, вона намагається автоматизувати завдання, які може реалізовувати зорова система людини.

Місії комп'ютерного зору містять в собі методи отримання, виділення, аналізу та розуміння цифрових зображень та вилучення багатовимірних матеріали з реального світу для отримання числової або символічної інформації, наприклад, у формі рішень [12]. Розуміння в цьому контексті означає перетворення зорових зображень в опису світу, які здатні взаємодіяти з другими розумовими процесами та викликати відповідні взаємодії. Дане розуміння зображення можна вивчати як відділення символічної відомості від даних зображення з застосуванням моделей, розроблених за використанням геометрії, фізики, статистики і теорії навчання.

Машинне бачення — це технологія та методи, які застосовуються для використання автоматизованої інспекції і аналізу на базі зображень для таких приложень, як автоматизована інспекція, управління процесом та керівництво роботами, на загальних засадах, в промисловості. Автоматичне визначення образів — це термін, що охоплює величезну кількість технологій, софтових і апаратних продуктів, скомпонованих систем, дій, методів та знань. Машинне розпізнавання як системну інженерну навчальний предмет можна вважати винятковою від комп'ютерного зору, формою комп'ютерних наук. Вона намагається включати існуючі технології по- новому та використовувати їх для розв'язання реальних світових проблем. Даний термін є розповсюджений для цих функцій в середовищах виробничої автоматизації, але також застосовується для різних функцій в інших середовищах, таких як забезпечення безпеки та керування транспортними засобами.

Як наукова дисципліна, машинне бачення відноситься до теорії штучних систем, які отримують інформацію з зображень [13]. Візуальний контент може приймати різні форми, наприклад відео послідовності, види із кількох камер чи багатовимірні дані з апарат для медичного сканування. В якості технологічної сфери комп'ютерний зір робить спроби для реалізації свої теорії та моделі для розробки систем комп'ютерного зору.

Сфери застосування варіюються від таких місій, як виробничі системи машинного зору, наприклад, контролюють прискорення виробів на виробничій

лінії, до аналізу в області штучного інтелекту та комп'ютерів чи роботів, здатних зрозуміти навколишній світ. Області комп'ютерного зору та машинного зору мають значну схожість. Комп'ютерний зір охоплює головну технологію автоматичного дослідження зображень, яка вживається в багатьох областях. Під машинним зором у більшості випадків розуміється процес об'єднання автоматизованого дослідження зображень з іншими методами та технологіями для гарантування автоматичного контролю та управління роботом в виробничих додатках [14]. У великій кількості додатків комп'ютерного зору інформаційні машини попередньо запрограмовані для розв'язання конкретного місії, але методи, розроблені на навчання, в сучасний період стають все більш поширеними. Приклади використання комп'ютерного зору включають системи для:

- автоматичне керування, наприклад, в промислових додатках;
- допомога особам в задачах розпізнавання, наприклад, система ідентифікації типів;
- управління етапами, наприклад, промислового робота;
- виявлення подій, наприклад, для візуального спостереження або підрахунку персоналу;
- взаємодія, також, як вхід для систем для взаємодії комп'ютера та працівників;
- моделювання об'єктів чи середовищ, наприклад, обробка медичних зображень чи топографічне моделювання;
- навігація, за допомогою автономного транспортного засобу чи мобільного робота;
- організація інформації, наприклад, для індексації баз даних зображень і послідовностей зображень.

Одна з найбільш головних областей застосування — медичний автоматичне розпізнавання образів чи обробка медичних зображень, що представлений отриманням інформації із даних зображень для визначення хвороби пацієнта. Прикладом цього є розпізнавання пухлин, артеріосклерозу чи

інших злякисних трансформації; виміру параметрів органів, кровотоку і таке інше є ще подібним прикладом. Він також забезпечує медичні дослідження, даруючи нову інформацію: наприклад, про будову мозку чи про якість медичної допомоги. Використання комп'ютерного зору в медичній сфері також включає покращення зображень, інтерпретованих персонажами, наприклад ультразвукові фотографії чи рентгенівські знімки, для обмеження впливу шуму.

Наступна область використання в комп'ютерному зорі знаходиться в виробничих, інколи званої машинним зором, де знання витягуються з метою підтримки промислового процесу [15]. Одним із прикладів є перевірка якості, при якому деталі чи фінальні продукти автоматично оцінюються на наявність дефектів. Відмінним прикладом є дослідження положення та орієнтації деталей, які мають бути підняті рукою механізованого робота. Машинний зір також широко застосовується в сільськогосподарській сфері для видалення небажаних продуктів з сипучого матеріалу, даний процес називається оптичним сортуванням.

Військові додатки, найімовірніше, є однією з найбільших сфер для комп'ютерного зору. Зрозумілим прикладами є виявлення противників солдатів чи транспортних засобів та наведення ракет. Більш високоякісні системи наведення снарядів направляють ракету в зону, а не в конкретну ціль, та вибір цілі здійснюється, коли ракета досягає зони, на основі інформації, отриманої на місці. Сучасні військові підходи, такі як «освіченість на поле бою», розуміють, що різні сенсорні елементи, включаючи датчики відображення, надають багатий асортимент інформації про бойову ситуацію, яка може застосовуватися для допомоги в стратегічних рішеннях. В даному випадку автоматична обробка мультимедійної інформації використовується для зниження складності та об'єднання інформації від декількох сенсорів для підвищення якості аналізу.

Однією з сучасних областей застосування є автоматизовані транспортні засоби, які мають в собі заглибні апарати, терестріальні транспортні засоби (невеликі механізми з колесами, легкові чи вантажні автомобілі), повітряні апарати та безпілотні літальні апарати (БПЛА). Ступінь автономності варіюється

від загалом автономних (безпілотних) транспортних механізмів до транспортних засобів, де структури на основі комп'ютерного зору допомагають водію або пілоту в різних ситуаціях. Загалом автономні транспортні засоби часто використовують комп'ютерний зір для маршрутизації, наприклад, для того, щоб отримати інформацію, де вона знаходиться, чи для складання карти навколишньої області (SLAM) та для виявлення перешкод. Дана технологія також може бути використана для знаходження певних подій конкретної місії, наприклад, БПЛА, які розшукують лісові пожежі. Прикладами допоміжних систем є системи оповіщення про перешкоди в транспортах та системи автономної посадки надземних транспортних засобів. Кілька автовиробників продемонстрували системи самостійного водіння автомобілів, але дана технологія все ще не набрала рівня, на якому може бути відображеною на ринку. Є безліч прикладів військових самостійних транспортних засобів, розпочинаючи від новітніх ракет та закінчуючи БПЛА для розвідувальних завдань або наведення снарядів. Освоєння космосу вже відбувається за допомогою автономних пристроїв, що застосовують комп'ютерний зір, наприклад, Mars Exploration Rover розроблено NASA та EchoMars Rover створено ESA.

1.3 Аналіз способів пошуку та виділення обличчя

Знаходження обличчя означає побудову певного набору правил, яким обов'язково відповідати фрагмент зображення, з метою, щоб бути визнаним людським ликом, наприклад: людина симетрична, риси лиця (ніс, очі, рот) відрізняються від епідермального покриву по яскравості, характеристики обличчя розташовані цілком індивідуальним чином. Також сюди можливо віднести знаходження за допомогою шаблонів, введених розробником. Знаходження обличчя за допомогою шаблону базується в перевірці всіх з областей фотографії на відповідність заданому формату.

З усього різноманіття існуючих методів виявлення обличчя можна виокремити декілька актуальних методів, які використовують уваги. Розглянемо особливості, переваги та недоліки кожного.

Метод Віоли-Джонса був запропонований Полом Віолою та Майклом Джонсом в 2001 році та став першим підходом, який демонструє якісні результати при аналізі зображень в реальному часі.

AdaBoost (Adaptive Boosting) — це метод підсилення класифікаторів використовуючи об'єднання їх в «комітет». Початково був запропонований Йоавом Фройндом та Робертом Шапіре. Цей метод може застосовуватися в поєднанні з деякими алгоритмами класифікації для підвищення ефективності. Цей алгоритм є адаптивним в тому сенсі, що всі наступний комітет класифікаторів створюються по об'єктам, неправильно класифікованим попередніми комітетами. Регулярно всього застосовується в поєднанні з іншими методами класифікації для їхньої підсилення (наприклад, як було описано раніше, в алгоритмі Віоли-Джонса).

Працює AdaBoost ось так. Кожну ознака використовується до кожного зображення. Селекціонується ознаки з найменшою набором різних помилок та похибок [16].

На початку тестовим зображення надається однакова вага, після кожної неправильної класифікації параметр вага зображень збільшується. Даний процес проходить до тих пір, поки не отримають потрібну точність, або досягнутий зазначеного коефіцієнта помилок, чи відібрана певна кількість характеристик.

Після даної маніпуляції кількість ознак значно зменшується, але в фотографіях, в загальних випадках, обличчя займає незначну область, і щоб не включати всі ознаки, які недоцільні для даної області, добавили поняття каскад класифікатор. На даний момент ознаки групуються по етапах. Перші стадії містять незначну кількість ознак, в кожній подальшій стадії їх кількість буде розширюватися. Область, яка пройшла всі попередні стадії, є обличчям.

SNoW (Sparse Network of Winnows) — метод розпізнавання обличчя, який являється собою двошарову мережу, перший шар якої формується з вузлів, всі з яких відповідають певній характеристиці вхідного малюнку, вихідний в свою чергу формується всього з двох нейронів, кожний з яких відповідає ідентифікації класам зображення (створює 1 при наявності певної особливості та 0 в разі її

відсутності в зображення), вихідний же формується всього з двох нейронів, кожен з яких відповідає ідентифікації класам зображень («обличчя» чи «не обличчя»). В якості характеристик в даному методі застосовується SMQT (Successive Mean Quantization Transform) характеристика, прাপори рівності деяким величинам середнього значення та дисперсії в кожному з прямокутоподібний фрагментів зображення з такими вимірами 2 на 2, 1 на 1, 4 на 4 та 10 на 10 (всі зображення містять параметри 20 на 20 пікселів). При реалізації класифікації на вхідні нейрони подається інформація про наявності певних характеристик в оброблюваному малюнку. Вузли вихідного шару розраховують лінійну комбінацію сигналів, які виникають вхідними вузлами. Коефіцієнти прямолінійної комбінації задаються вагами серед вхідними вузлами. Дані маніпуляції дозволяють вилучити з локальної зони зображення складову, яка самостійна від освітленості. Воно базується в квантуванні області рисунку з порогом квантування тотожнім середньому значенню пікселів, які знаходяться в цій зоні.

SNoW спеціально створено для випадків класифікації, коли перспективне число характеристик об'єктів, необхідних для класифікації може бути вельми великим, але невідомо заздалегідь. Розріджена побудова мережі дозволяє застосовувати величезну кількість характеристик зображення в якості вхідних параметрів, оскільки в процесі тренування усі несуттєві параметри відкидаються, і не уповільнюють зрештою, працездатність класифікатора.

Нейромережеві алгоритми включають в себе цілий клас методів. Основна ідея — впорядкування перетворень сигналу паралельно обробляючим функціональними елементами, нейронами. Значення процесу навчання таких мереж обмежується до зменшення середньоквадратичної похибки. Системи виявлення об'єктів на зображенні, засновані на нейронних мережах, застосовують ієрархічну структура. Спочатку вектор характеристик оброблюється грубою мережею із високим рівнем помилок другого порядку, далі, за умови, що вектор не був систематизованим як не об'єкт, рішення модифікується більш точною та ефективнішою мережею.

Структура багат шарової нейронної мережі (БНМ) складається із послідовно об'єднаних шарів, де нейрон кожного оболонки своїми входами пов'язаний із всіма нейронами попереднього шару, а виходами до наступного. Нейронна мережа із двома визначальними шарами може із будь якою точністю наближати до прийнятного значення будь яку багатовимірну функцію. Нейронна мережа (НМ) із одним ключовим шаром здатна створювати лінійні розділяючі поверхні, що значно звужують коло місій, які вони вирішують, особливо, така мережа не зможе розв'язати завдання типу «виключне чи». Нейронна мережа із нелінійної функцією активації та двома вирішальними шарами дозволяє створювати будь-які опуклі області в області рішень, а з трьома ключовими шарами — зона будь-якої важкості, в тому числі та неопуклої. При цьому багат шарова нейронна мережа зберігає своєю узагальнюючою здатністю. Навчаються БНМ за посередництвом алгоритму зворотного поширення відхилення, що є методом градієнтного зниження в просторі ваг із метою мінімізації сумарної похибки мережі. При цьому помилки (детальніше, величини корекції ваг) розповсюджується в зворотному напрямку від входу до виходу, крізь ваги, що сполучають нейрони. Найпростіше використання одношарової нейронної мережі (відомою як автоасоціативною пам'яттю) закладається в навчанні мережі регенерації зображень, які подаються на вхід. Посилаючи на вхід тестове зображення та обчислюючи якість реконструйованого рисунку, можна оцінити, наскільки мережа визнала вхідне зображення. Позитивні характеристики цього методу полягають у тому, що мережа може реконструювати спотворені та зашумлені рисунки, але для більш серйозних завдань він не підходить. БНМ так же застосовується для безпосередньої систематизація зображень — на вхід подається чи саме малюнок в будь-якому форматі, чи набір раніше витягнутих ключових параметрів зображення, на виході нейрон із максимальною активністю показує приналежність до розпізнаного типу.

Метод опорних векторів створений на тому, що шукається лінійний розбиття класів. Мета навчання більшості класифікаторів — мінімізувати

похибку класифікації на тренувальному сеті (звану емпіричним загрозам). На відміну від них, за допомогою алгоритму опорних векторів можна створити класифікатор, мінімізуючий верхню межу очікуваної помилки класифікації (в тому числі та для невідомих екземплярів, що не входили в тренувальний асортимент).

Застосування алгоритму опорних векторів до місії виявлення особи полягає у знаходженні гіперплощин в просторі ознак, що виокремлює клас зображень обличчя від інших «не обличчя». Можливість лінійного розподіл таких складних класів, як рисунків обличчя і «не обличчя» є мало ймовірною. Одна систематизація за допомогою опорних векторів гарантує використовувати апарат ядерних функцій для таємного проектування векторів-ознак в простір потенційно більш високої розмірності (не вище, ніж простір рисунку), в якому класи можуть з'ясувати лінійно розподілені. Неявне творення за допомогою ядерних функцій не веде до ускладнення обчислень, що дозволяє успішно використовувати лінійний розподільник для лінійно нероздільних типів.

В результаті роботи системи виділення персон можуть виникати два типи похибок: не виділення обличчя і хибне виявлення (виділення елемента, який обличчям не є). В зв'язку з наявністю двох класів помилок, існує два ключових параметри, що характеризують продуктивність роботи алгоритмів виявлення лиця: рівень виявлення, що вказує відсоток виявлених обличчя, та рівень невірною виділення, ідентичний загальному числу помилкових виявлень на усьому тестовому сеті.

Суть методу полягає у знаходженні гіперплощини, що поділяє два класи. При цьому із усіх можливих гіперплощин, які розділяють два класи, необхідно обрати таку гіперплощину, довжина до якої від кожного типу була максимальною. І ця гіперплощина називається абсолютно розділова гіперплощина, а відповідний даній лінійний класифікатор має називається максимально розділовим класифікатором.

Одне й те ж обличчя, виділене різними методами, має різний розмір. Так метод на базі бустинга виділяє обличчя повністю, захвачуючи лоб, підборіддя та

щоки. Метод на базі бустингу не є характерними алгоритмами для виділення лиць. Вони, скоріше, використовуються для удосконалення дискримінативних моделей, наприклад, для розпізнавання об'єктів на рисунках, а не для виділення самого обличчя. Метод на базі Snow та опорних векторів виокремлює тільки очі, ніс та рот, причому метод на базі опорних векторів виокремлює обличчя вузьким вікном. Дані відмінності викликані тим, що у процесі створення методів використовувалися різні набори тренувальних зображень для побудови розподільника.

Методи основані на використанні нейронних мереж надають одні із найкращих результатів розпізнавання обличь. Така успішність досягається за допомогою застосування згорткових нейронних мереж, що ґрунтуються на архітектурних рішеннях когнітрон і неокгнітрон. Даний алгоритм характеризується стійкістю до перемін масштабу, зсувів, поворотам, переміні ракурсу і таке інше. Результати тестувань показують ступінь успішного розпізнавання і правильності розв'язку на рівні 96% при існуванні спотворюючих факторів.

Головним недоліками даного алгоритму є необхідність навчання нейронної мережі кожного разу коли необхідно включити нову особу для ідентифікації. Час тренування збільшується відносно кількості осіб у базі обличь. Хоча недолік необхідності навчання мережі іноді є значущим, навіть при це даний метод є одним із найефективніших на даний час.

Алгоритм розпізнавання осіб можна поділити на дві підгрупи. Перша підгрупа — методи, створені на значеннях пікселів, та методи, засновані на певних точках. Розглянемо ключові методи кожної із цих підгруп.

Алгоритми, засновані на значеннях яскравості графічних точок. Назва даної групи алгоритмів означає, що для ідентифікації виявлених обличчя використовується лише колір або яскравість пікселів на рисунках. Найпростішим схожим методом є зіставлення, в якому мірою подібності є відстань між векторами інтенсивності пікселів зображень. Однак даний алгоритм абсолютно не стійкий до перемін освітленості, положення обличь,

масштабування. Більш того, даний підхід має високу розрахункову складність та абсолютно не придатний для ідентифікації у реальному часі. Тому, частенько використовуються методи, які застосовується перехід векторного опису рисунку в простору із меншою розмірністю, в яких зіставлення набагато результативніше.

Eigenfaces — метод, запропонований Метью Терком і Алексом Пентланд, який отримав широку популярність в якості першого успішного алгоритму розпізнавання осіб. Ключовою ідеєю алгоритму є застосування алгоритму головних компонент для пошуку векторів, найкращим чином характеризують зображення осіб. Використовуючи цей алгоритм можна виявити різні модифікації в навчальній вибірці зображень обличь та описати цю зміну в базисі пари ортогональних векторів, які називаються індивідуальними (eigenface). Обличчя, мають ефективну властивість: зображення, яке узгоджується кожному вектору має форму обличь. Обчислення головних компонент резюмується до обчислення власних векторів та власних значень коваріантної матриці, яка обчислюється із зображення. Отриманий один раз на тренувальній вибірці зображень обличь набір власних векторів застосовується для кодування всіх інших рисунків осіб, які представляються зваженої поєднанням цих власних векторів. Застосовуючи обмежену кількість своїх векторів можна отримати кратку апроксимацію вхідному зображенню персони, яку потім можна зберігати у базі даних в вигляді вектору коефіцієнтів, який використовується одночасно ключем пошуку у базі даних осіб. Сума ключових компонент помножених на узгоджені власні вектори являється реконструкцією рисунку.

Для кожного рисунку обличчя обчислюється його ключові компоненти. Зазвичай це від 5 до 200 ключових компонентів. Процес розпізнавання ґрунтується на зіставленні головних компонент невідомого зображення із компонентами всіх відомих рисунків.

Метод власних обличь вимагає для свого застосування ідеалізування критерії таких як, єдині параметри яскравість, нейтральний вираз обличчя, наявність завад в якості окулярів та бороди. При недотриманні цих умов ключові компоненти не будуть віддзеркалювати міжкласові варіації. Наприклад, при

різноманітних умовах освітленості метод індивідуальних обличь майже не застосовується, внаслідок того, що перші головні компоненти переважно показують зміни освітленості, та порівняння видає зображення, яке має подібний рівень освітленості. При додержанні ідеалізованих умов точності виявлення з використанням алгоритму досягає більше 90%.

Розрахунок набору власних векторів виокремлюється високою трудомісткістю. Один із методів — верстка зображень по рядках і стовпчикам; в такій формі представлення рисунку має на порядок менший об'єм, обчислення та розпізнавання проходить швидше, але реабілітувати початкове зображення неможливо.

Алгоритм головних компонент добре відзначився себе в практичних додатках. Але, в тих ситуаціях, коли на зображенні персони присутні значні зміни в яскравості або виразі обличчя, продуктивності методу значно падає.

Ідея алгоритму складається з представлення рисунку обличчя у вигляді набору векторів ключових компонент рисунку.

Fisherfaces — методу, в якому на відміну від алгоритму eigenfaces використовується лінійний дискримінантний дослідження, а саме лінійний дискримінант Фішера. Дія методу заснована на пошуку проекції інформації, при якій класи зображень обличь максимально відокремлені. При використанні алгоритму головних компонент здійснюється максимізація розкиду інформації по всій базі обличь. Ця відмінність дозволяє розв'язати проблему високої вразливості до змін освітлення .

Алгоритм головних компонент, який являється ключовим методу Eigenfaces, знаходить лінійну поєднання ознак, яка максимізує спільну дисперсію даних. Це доволі ефективний спосіб представлення інформації, але він не узгоджує класи об'єктів, та може бути втрачено безліч відмінної інформації через відмови компонент, які не перейшли базис. Наприклад, коли в наборі є фотографія з різним яскравістю, та оскільки відомо, що елементи, визначені за допомогою алгоритму головних компонент, не завжди мають в собі всю відмінну відомості, тому зразки різних типів змішались один з одним, та подальша

класифікація стає недоступною. Цю проблему рішає метод лінійного дискримінантного дослідження, який є основою алгоритму Fisherfaces.

Лінійний дискримінантний аналіз — алгоритм статистики машинного тренування, який застосовується для виявлення лінійних комбінацій признаків, які найкращим чином розкладають два або більше класи об'єктів або подій (тобто об'єкти одного та того ж класу мають витягуватись, як можна вблизі один до одного у просторі, та при цьому відбувається оптимізація відстані між класами).

Даний метод краще розпізнає при різній яскравості, чим EigenFaces, тільки тренування має відбуватись крім цього при різних умовах освітленості.

Алгоритми, засновані на характерних точках. Зазначена група методів, розбіжна від попередньої, використовує ключові точки та їх координати на рисунку, а не оцінює яскравість бітові елементи. Такими характерними точками можуть бути, для ілюстрації, центри очей, розміщення носа, лінія брів, вуха і т. д. До даного типу методів відносяться активні форми зовнішнього вигляду та активні моделі форми.

Дієві моделі зовнішнього зовнішності (Active Appearance Models, AAM) — це аналітичні моделі зображень, які шляхом різноманітного роду деформацій допускають підігнані під реальне рисунки. Даний тип моделей у двовимірному альтернативі було запропоновано Тімом Кутсом та Крісом Тейлором. Активна модель зовнішнього вигляду містить два види параметрів: параметри, пов'язані із формою (параметри форми), та параметри, взаємопов'язані зі статистичною імітацією пікселів зображення чи текстурою (характеристики зовнішнього вигляду). Перед застосуванням модель повинна бути натренованою на безлічі заздалегідь розміркованих зображень. Розмітка рисунків виробляється вручну. Завдяки застосуванню активної моделі наявність завад, як, наприклад, вуса, структура в частині випадків зможе зазначити відповідь, оскільки розходження значень на роздільних точках може не відображати на загальну відповідь програми. Спочатку здійснюється локалізація обличчя за допомогою маски обличчя. Потім виконується пересування обличчя, використовуючи виявлену

позиції і розмір за використання активного контуру моделі і відповідно з цієї моделі отримуються головні точки.

Динамічні моделі форми (Active Shape Models, ASM) розглядають статистичні зв'язки в взаємному розташуванні антропометричних пікселів. На кожному рисунку вибірки експерт розмічає локацію антропометричних пікселів. Для того щоб провести координати на всіх рисунках до єдиної системи в основному виконується узагальнений покрутом дослідження, в результаті якого усі точки приводяться до одного масштабу та центруються. Далі для всіх з набору образів розраховується середня форма та матриця коваріації. На основі матриці кореляції обчислюються свої вектори, які необхідні сортуються у порядку спадання взаємопов'язаності їх власних значень.

Однак необхідно зазначити, що схожі моделі спочатку назначені не для ідентифікації, а для точного знаходження характерних точок в зображеннях обличчя. Їх локалізація дасть виконати процедуру вирівнювання обличчя вибірки та приведення їх до однієї бази координат для більш ефективного розпізнавання іншими методами. Часто для цих цілей застосовується невелика кількість пікселів, що дозволяє прискорити розпізнавання. Для задач розпізнавання, навпаки, необхідна велика кількість характерних пікселів, що збільшить точність класифікації та знизити швидкість роботи програми.

У результаті дослідження можна відзначити, що оптимальним алгоритмом для обробки у реальному часі має метод із використанням нейронних прикладів. В наступних розділах даний алгоритм буде описаний більш уважно, а також буде наведено аналіз продуктивності різних модифікацій алгоритму, яке дозволить знайти найкращу для вирішення поставленої місії модифікацію із точки зору співвідношення швидкодії роботи та точності класифікації.

1.4 Нейромережа глибокого навчання для розпізнавання людей

Нейромережа глибокого навчання, також відома як глибока нейромережа, є типом штучної нейромережі, яка складається з багатьох шарів нейронів, які намагаються моделювати складні залежності у вхідних даних. Ця концепція

базується на імітації функціонування людського мозку, де інформація обробляється через багато рівнів обробки.

Глибокі нейромережі мають багато прихованих шарів між вхідним і вихідним шарами, які називаються «глибиною» нейромережі. Ці приховані шари дозволяють моделі автоматично визначати важливі ознаки або патерни у вхідних даних, що робить їх здатними вирішувати складні завдання, такі як розпізнавання об'єктів у зображеннях, обробка природної мови, генерація тексту та багато інших завдань машинного навчання.

Глибоке навчання вимагає великої кількості даних для тренування та вагомих обчислювальних матеріалів, таких як графічні процесори (GPU) чи спеціалізовані тензорні процесори (TPU). Популярні архітектури глибоких нейромереж включають у себе звичайні нейронні мережі (Feedforward Neural Networks), та згорткові нейронні мережі (Convolutional Neural Networks) для обробки зображень та рекурентні нейронні мережі (Recurrent Neural Networks) для роботи з послідовними даними.

Глибокі нейромережі здійснюють значний вплив на багато сфер життя, включаючи технології розпізнавання образів, автономних автомобілів, системи рекомендацій, медичинську діагностику та багато інших.

Мережі часто реалізуються в вигляді комп'ютерних системах, хоча випускається все більша та більша кількість інтегральна схема, що реалізують нейронні мережі інструментальним шляхом. Ключова властивість мереж — здатність до навчання. Глибоке навчання — набір алгоритмів, які намагаються моделювати високорівневі абстракції у даних, використовуючи архітектури, що складаються із безлічі нелінійних перетворень. Глибока нейронна мережа (Deep Neural Network, DNN) це синтетична нейромережа із декількома прихованими шарами. Схожими до звичайних синтетичних нейромереж, глибокі нейронні мережі здатні моделювати складні нелінійні взаємини між елементами. В процесі навчання глибокої нейронної мережі здобута модель намагається представити матеріали у вигляді комбінації не складних примітивів (наприклад, в задачі розпізнавання людей такими примітивами можуть бути елементи обличчя: ніс,

очі, рот та інші). Додаткові шари дозволяють створювати абстракції все більш високих ступенів, що і дає будувати моделі для ідентифікувати складних об'єктів навколишнього світу.

Як правило, глибинні мережі створюються як мережі прямого поширення. Але останні дослідження показали, як треба застосувати техніку глибинного тренування для рекурентних нейронних мереж. Згорткові комп'ютерні нейронні мережі використовуються у області машинного зору, і цей метод показав себе як ефективний. Так же згорткові нейронні мережі були використані для розпізнавання мови.

Тренування глибинних нейронних мереж може бути реалізовано за допомогою звичайного методу зворотного поширення помилки. Є велика кількість модифікацій цього алгоритму. Таким чином, можливо бути використано кілька правил конфігурація ваг.

До двох ключових проблем глибоких нейронних мереж класифікують ті ж самі проблеми, що появляються й при навчанні нормальних нейронних мереж: час тренування й перенавчання.

Глибокі структури більш схильні до перенавчання, так як, маючи більше шарів, що забезпечують моделювати високо рівневі абстракції, мережа може «натренуватися» рідкісні ситуації. В цьому випадку можуть сприяти різні види регуляризації. Один із можливих алгоритмів регуляризації (dropout) припускає випадковим чином відсутні вузли під час навчання. В деяких випадках це асистує менше запам'ятовувати рідкісні залежності у тренувальних даних.

Через простоту реалізації та хорошу збіжність для тренування глибоких нейронних мереж зазвичай використовується алгоритм зворотного поширення помилки та градієнтний спуск. Однак, при тренуванні глибоких структур виникає декілька проблем, які особливо необхідні при оптимізації функцій в просторі великої розмірності: кількість вираховувальних елементів, початкові умови для параметрів мережі, а також описана вище стала величина регулювання величини кроку.

Крім того, метод стохастичного градієнтного спуску славетні власною проблемою зникаючого градієнта (vanishing gradient), яка є в ослабленні градієнта, а значить та швидкості навчання у міру поглиблення від остаточних шарів мережі до старту мережі. Через це нижні шари мережі дуже погано тренуються. Проте останнім часом є схильність замість функції активації нейронів мережі виду сигмоїда у глибоких мережах використовувати нелінійність типу ReLU (Rectified Linear Unit), функцію яких можна розписати як $\max(0, x)$. Глибока мережа із таким видом функції активації не містить проблеми ослаблення градієнта та добре навчається градієнтним спуском. Лише за умов великих розмірностей з повним перебором всіх комбінацій значень ваг вона є непрактичною. виправлена лінійна функція активації чи скорочено ReLU є кусково-лінійною функцією, яка призводить вхідний сигнал безпосередньо, якщо він позитивний, інакше — нуль. Вона буде функцією активації за замовчуванням для чималого типів нейронних мереж, оскільки приклади, яка використовує її, легше тренувати та часто досягає кращої ефективності.

Моделі глибокого навчання формуються з різноманітних глибинних мереж. Поміж ними глибокі короткі мережі (DBN), глибокі нейронні мережі (DNN), згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN) має контрольованими моделями навчання. Моделі тренування без нагляду це автокодери, мають ліміти машини Больцмана (RBM) і генеративні змагальні мережі (GAN). З 2015 року обсяг досліджень IDS, що основані на глибокому тренуванні почала швидко зростати, що відбувається та сьогодні. Моделі глибокого тренування безпосередньо вивчають представлення функцій з вихідних даних, таких як рисунку та тексти, без необхідності мануальної інженерії особливостей. Останніми роками глибоке тренування досягло максимальних висот в різних сферах застосування. Ця нова сфера машинного навчання швидко підвищується та застосовується в більшості традиційних галузей додатків, а також в деяких нових галузях, які надають більше спроможностей. Таким чином, алгоритми глибокого навчання можуть реалізовуватися наскрізними способами. Для великих сетів даних методи

глибокого тренування мають величезне преимущество перед неглибокими прикладами. При вивченні глибокого тренування основними акцентами є мережева структура, вибір гіперпараметрів і стратегія оптимізації. Такі самі види моделей машинного тренування можуть мати різноманітні обмеження, параметри чи потребувати певної швидкості тренування для різних типів даних. Ці ваги називаються гіперпараметрами та їх слід підбирати, щоб модель могла оптимально розв'язати завдання навчання. Порівняння різноманітних моделей глибокого навчання вказано в таблиці 1.1.

Таблиця 1.1 — Порівняння різноманітних моделей глибокого навчання

Алгоритми	Відповідні типи даних	Під наглядом або без нагляду	Функції
Автокодер	Необроблені дані; Вектори характеристик	Без нагляду	Особливість зменшення; Шумить
RBM	Вектори характеристик	Без нагляду	Особливість зменшення;
DBN	Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
DNN	Вектори характеристик	Під наглядом	Видобуток особливостей,
CNN	Необроблені дані; Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
RNN	Необроблені дані; Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація

Глибоке навчання — це сфера машинного навчання. Ефекти моделей глибокого тренування, перевершують ефекти традиційних алгоритмів машинного навчання в більшості сценаріїв застосування. Відхилення між поверхневими моделями і глибокими моделями у основному відображаються у наступних аспектах.

Довго тривалість роботи. Час бігу містить як час тренувань, так та час випробувань. Через велику складність глибоких моделей та час їх тренування, час випробувань набагато триваліший, ніж в неглибоких моделях.

Кількість параметрів. Є два види параметрів: параметри, які можна дізнатись, і гіперпараметри. Ваги, що підлягають вивченню, розраховується на етапі навчання, а гіперпараметри вписуються вручну перед стартом навчання. Параметри і гіперпараметри, що вивчаються, в глибоких моделях значно випереджають такі, як в поверхневих моделях. Можна розробити висновок, що навчання і оптимізація глибоких моделей захоплює більше часу.

Демонстрація функції. Вхідні дані до звичайних моделей машинного тренування — це вектор функцій, а розробка особливостей — головний крок. Моделі ж глибокого тренування здатні вивчати подання об'єктів з необроблених даних та не залежать від інженерних можливостей. Алгоритми глибокого навчання можуть реалізовуватися наскрізним способом, даючи їм видатну перевагу перед звичайними методами машинного тренування.

Навчальна здатність. Структури прикладів глибокого навчання складні та містять величезну кількість ваг. Отже, моделі глибокого тренування мають сильніші здібності до адаптації, ніж моделі із поверхневим вивченням. Однак моделі глибокого тренування також стикаються із вищим ризиком переобладнання, потребують набагато більшого обсягу наборів даних для навчання. Отже ефект від моделей глибокого тренування кращий.

Моделі глибокого тренування — це чорні скриньки, висновки яких майже не піддаються тлумаченню. Це є критичним моментом в глибокому тренування. Однак деякі традиційні методи глибокого тренування, такі як дерево рішень і наївний Байєс, мають інтенсивну інтерпретацію.

Моделі глибокого тренування досягли значних успіхів у аналізі великих баз даних. Але їх результати не є досконалими для малих чи незбалансованих наборів інформації. Недоліком є суперечливі методів до навчання, що може поліпшити точність виявлення на малих наборах інформації.

У даному розділі було проведено аналіз алгоритмів, які застосовуються при обробці відео потоку з метою пошуку обличчя людини та ідентифікації особи за отриманим зображенням. Алгоритм з використанням нейромереж з глибоким навчанням є найбільш кращим, оскільки є можливість тренування. Використовуючи цей алгоритм в системах виявлення та розпізнавання особи, можна змінювати характеристики та послідовність пошуку, що реалізується у конкретній системі розпізнавання, проводячи навчання цієї системи на конкретних прикладах.

2 РОЗРОБКА НЕЙРОМЕРЕЖЕВОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ

У даному розділі розглянуті етапи формування пошуку обличчя людей у відео потоці, стадії підготовки та обробки даних по розпізнаванню обличчя та виконана розробка архітектури нейронної мережі

2.1 Етапи формування пошуку обличчя людей у відео потоці

Існують такі поширені методи розпізнавання:

- геометричний алгоритм;
- алгоритм головних компонент;
- нейромережеві алгоритми;
- фільтр Габора;
- схованої марковської моделі;
- алгоритм Віюлі-Джонса;
- вейвлет- модернізація.

При розв'язанні задачі розпізнавання персон з'являються дві проблеми. Першою проблемою є те, що будь-який рисунок це лише масив точок. Проте один піксель не містить істотного значення, адже модифікація його інтенсивності не вплине на результат ідентифікації. Тому подання зображень в вигляді набору пікселів є надлишковим та не ефективним. Отже, щоб ідентифікації було результативним необхідно використовувати зручний і ефективний формат для роботи з кадрами. На сьогоднішній день є можливість стиснути рисунок за допомогою безлічі методів, але результат ущільнення є не зручним виглядом для класифікації персон на зображеннях. Для розв'язання задачі розпізнавання людей не потрібно надмірної кількості даних. Це пов'язано в першу чергу із тим, що потрібно визначити, яка персона із словника найбільше схожа на персону, яка на вхідному зображенні. Наступна проблема полягає у тому, що потрібно враховувати різноманітні зовнішні фактори, як-от: освітлення, поза, емоції і таке інше.

Розглянемо приклад, у якому всі зображення мають розмір 100 на 100

пікселів із 256 відтінками сірого. Постановку місії розпізнавання персон можна сформулювати по різноманітному. Розглянемо інший варіант, що є деякий набір даних, яка налічує 400 зображень (по 10 фотографій для 40 персон при різних випадках). Місію розпізнавання осіб можна визначити наступним чином: є деяке нове зображення у форматі, що було розписано вище. У результаті вирішення поставленої задачі очікується наступна із відповідей:

- рисунок не є обличчям;
- фотографія є обличчям певної людини із словника;
- зображення має обличчя, але його немає у словнику.

В цьому випадку його потрібно додати до бібліотеки.

Як правило, спочатку потрібно виконати попередню обробку фотографії перед операцією ідентифікації, що дозволяє підвищити ефективність та якість розпізнавання шуканих чи досліджуваних матеріалів. Методи попередньої обробки зв'язані від завдань, які приймаються перед системою та можуть включати: виокремлення інформативних фрагментів, їх масштабування, отримання 3-мірних рисунків, покращення просторового дозволу, зростання контрастного дозволу, зростання якості зображень. Підсумовуючи, можливо, виділити наступні типи попередньої обробки рисунків:

- корекція контрастності і яскравості рисунку;
- покращення просторового дозволу;
- вирівнювання освітленості рисунку;
- гістограми освітленості.

Рисунку, що вводяться у комп'ютер, зазвичай є мало контрастними. Нормальний слабкий контраст є наслідком широкого діапазону освітленості, що відтворюються у поєднанні із нелінійною характеристикою поширення рівнів. Також на якість фотографії впливає характер зміни освітленості пікселів від мінімального до максимального значення, адже прямолінійна функція зміни інтенсивності точок є оптимальною. Якщо характеристика рисунку буде увігнутою, тоді фотографія більш темніша, а при опуклій характеристиці є більш світлою. У обох випадках ознаки предметів можуть бути спотворені та

недостатньо добре розпізнано. Корекція гистограми освітленості істотно покращує якість рисунку.

Мала контрастність дозволяє бути обумовлена й тим, що різноманіття функції яскравості точок на зображенні більш менше допустимого діапазону шкали освітленості. В цьому випадку контрастність рисунку підвищується шляхом збільшення динамічного діапазону освітленості на всю шкалу використовуючи лінійного по елементного модифікація.

Інверсія вхідного рисунку є іншим способом зміни яскравості палітри. Оскільки виділяти слабкі сигнали на мраковому фоні непросто, то протилежна форма подання рисунку має більш задовільному форматі для спостереження та візуальної розпізнавання.

Деякі місії обробки зображення пов'язані із перетворенням на півтонового рисунку (різні градації освітленості) в бінарне. Щоб зменшити інформаційну надмірність рисунку здійснюється змінення зображення у бінарне. У результаті залишається лише дані, які потрібні для вирішення місії розпізнавання. В бінарному рисунку повинні бути знешкоджені несуттєві признаки, наприклад, фон та збережені необхідні деталі, наприклад, обриси рисунку об'єктів.

У силу недостатньої експозиції розмах яскравості фотографії у цифровому вигляді може містити відмінності від діапазону освітленості вихідного рисунку. Виконати корекцію освітленості можливо за використанні двох методів. Перший спосіб передбачає, що рисунок лінійно відображається у діапазоні освітленості вихідного. При використанні другого способу потрібно виконати обмеження яскравості точок між максимальним та мінімальним пороговими значень. Присутність в зображенні найсвітліших та темних тонів створює враження необхідної контрастності. Надмірна контрастність приводить, що максимальні градації приводять до середніх тонів, адже більшість елементів рисунку знаходяться у середніх тонах, а зайва яскравість може спричинити розгублення цих елементів або ускладнити їх виокремлення.

У більшості випадків певні зони на зображенні бувають набагато темними, щоб на них можливо було щось розпізнати. Просте підвищення освітленості для

всього рисунку, може призвести, що світлі фрагменти можуть стати засвіченими. Щоб поліпшити вигляд зображення у таких випадках, використовують метод вирівнювання яскравості.

Яскравість можна вважати низькочастотним елементом, оскільки освітленість змінюється в просторі повільно, а фотографія — високочастотним сигналом, оскільки рисунок може містити невеликі деталі, що призводить до швидких модифікацій в його текстурі і конфігурації. Добуток цих елементів можна представити в вигляді результуючого імпульсу на первинному зображенні.

Головним завданням гомморфної обробки рисунку є нормалізація рівнів освітленості, а саме звуження їх активного діапазону і одночасному підвищенні його яскравості.

При обробці рисунку важливе значення мають методи рангової фільтрації, адже їх функціонал дає позбавлятися розмитих елементів. Поліпшення фокусу концентрації виконується за рахунок найкращого вибору двовимірної матриці n на n маски точок. Ранжирування значень інтенсивності здійснюється в межах маски, а центральній точці присвоюється значення, яке відповідає максимальному значенню по рівню.

Лінійна фільтрація дає можливість для удосконалення спотворень різного виду, що виникають внаслідок недосконалості апаратури, що створюють рисунки. Окрім того, за допомогою прямолінійні фільтрації здійснюється зменшення впливу флуктуаційних шумів і інших дефектів на рисунках у режимі підвищення яскравості мало контрастних деталей при посиленні масштабу зон інтересу.

Модифікація спотворень зображень у разі відсутності шумових елементів сигналу виконується шляхом інверсної фільтрації (деконволюції). Проте потрібно пам'ятати, що фільтри зворотного конволювання мають коефіцієнт підвищення дисперсії шумів, більший 1, та натомість можливо збільшитися зашумленість рисунку.

Перспективними вважаються нелінійні алгоритми фільтрації з

використанням частотних масок, які дозволяють знижувати вплив низькочастотних елементів сигналу зображення та посилити вплив високочастотних елементів, що підвищує просторову точність відображення деталей, описаних в спектрі фотографії більш високими просторовими частотами.

У зоні систем машинного зору і штучного інтелекту виділення обличчя та подальшої розпізнавання особи завжди перебувала у ряду найбільш пріоритетних місій для дослідників. Досить суттєвою проблемою у системах комп'ютерного зору є велика пере мінливість візуальних образів, яка пов'язана з змінами яскравості, забарвлення, масштабів, кутів спостереження. Проте найбільшою завадою для програм комп'ютерного зору є усунення двозначності, що виникає при створенні тривимірних об'єктів навколишнього світу на плоскі рисунки. Завдання виявлення елементів на зображенні ускладнюється через масштабний обсяг даних, що містяться на рисунку. Зображення може мати тисячі точок, кожен з яких може містити істотне значення, а дослідження кожного пікселя може призвести до тривалим процесом.

Набір оптимального методу опису об'єктів здатний розв'язати дану проблему. Опис елементу має його враховувати найбільш істотні характеристики та бути досить детальним, щоб виокремлювати даний об'єкт від інших об'єктів, що можуть існувати на зображенні.

При виокремлення людини і ідентифікації за рисунком можна виділити наступні етапи:

- перевірка присутності персони на аналізованій сцені;
- виокремлення фігури людини;
- виокремлення голови;
- аналіз ракурсу спостереження лиця (профіль);
- виокремлення особи;
- порівняння із еталонами та ідентифікація.

Залежно від характерних рис поставленої задачі — виконання окремих кроків методу можуть відрізнятися. При виконанні системи виявлення і

ідентифікації персони по зображенню потрібно здійснити фільтрацію величезного потоку вхідних даних і виокремлення істотних ознак, щоб мати задовільні результати виконання етапів алгоритму. Метод повинен вміти ефективно відрізати статичні елементи сцени, що недоцільно змінюються, працювати у різних умовах яскравості, виявляти людину під різними кутами спостереження. Для забезпечення необхідних можливостей методу необхідна фото з великою роздільною здатністю, що буде надавати кращу достовірність розпізнавання. Завдання визначення факту присутності персони на сцені, вимагає від методу певного рівня обізнаності. Це не повинна бути програма, що відреагує на факт зміни сцени. Метод виявлення персони не повинен здійснювати помилкові спрацювання при модифікаціях освітленості, руху тіней.

При виборі методу ідентифікації персони по зображенню необхідно відштовхуватися від положення його застосування. Наприклад, із місями розпізнавання у обмеженому приміщенні легко справляється багат шарова нейронна мережа. В той же час міся виявлення конкретної персони серед групи людей потребує застосування більш складних методів для зменшення випадкових реагувань. Скоріш за все, в цьому випадку необхідно багаторівнева система, що матиме безліч аналізаторів, які досліджуватимуть різні ознаки.

Типовий метод розпізнавання складається з наступних етапів:

- перетворення вихідного рисунку в стандартне представлення;
- виокремлення ключових характеристик;
- механізм розпізнавання.

2.2 Підготовка та обробка даних для розпізнавання обличчя

Для виявлення зображень обличчя у відео потоці запропоновано використовувати модель активної форми у вигляді ключових точок та форму контуру обличчя людини для її розпізнавання, а також використати гістограми направлених градієнтів.

Концепція гістограм напрямлених градієнтів закладається в тому, що замість того, щоб експлуатувати напрямок градієнту кожної точки окремо, ми

виокремлюємо їх в комірки. Для кожної комірки обчислюються усі напрямки градієнтів, а потім всі напрямки в комірці підсумовуються. Чим більший вектор, тим збільшується вага він має для комірки, а випадкові маленькі напрямки відкидаються. Тобто ця гістограма надає визначальну орієнтацію комірки. А всі комірки разом — уявлення про структуру рисунку загалом. Гістограма містить загальну подачу елементу, але дає при цьому шанс варіативності.

Лінійний алгоритм опорних векторів є двійковим лінійним класифікатором. Шаблон є представленням прикладів як точок в просторі, відображених таким чином, що приклади із окремих категорій виокремлені прогалиною, яка є як найбільшою. Нові приклади також відображаються в даному просторі, та на основі цього виконується передбачення про їхню приналежність до певного класу на основі того, в яку сторону вони впадають.

Відповідно, щоб знайти лице на фото, спочатку метод знаходить гістограми напрямлених градієнтів. Наступний крок, застосовуючи ковзке вікно, переходить по клітинам гістограм і перевіряє за допомогою прямолінійного методу опорних векторів або наявне у цьому вікні лице. Наприкінці методу вибирає із усіх знайдених позитивних фрагментів найкраще.

Для попередньої модифікації зображення пропонується використати алгоритми із застосуванням моделі активної форми. Фреймворк активної форми — це статична схема, що може як завгодно змінювати щоб відповідати об'єкту на оновленому рисунку. Форма складається із певної кількості пікселів, кожна з яких узгоджується за своє місце на елементі, у нашому випадку — на лиці. На рисунку 2.1 продемонстровано початкова модель активної форми лиця людини, яка містить в собі 68 точок [16].

Для того, щоб виявляти точки моделі на оновленому обличчі, необхідно навчити модель розшукувати найкращі позиції і виконувати співставлення із еталоном. Для розшуку точок застосовується відстань Махаланобіса, довжина у евклідовому просторі, що узагальнює значення евклідової відстані.

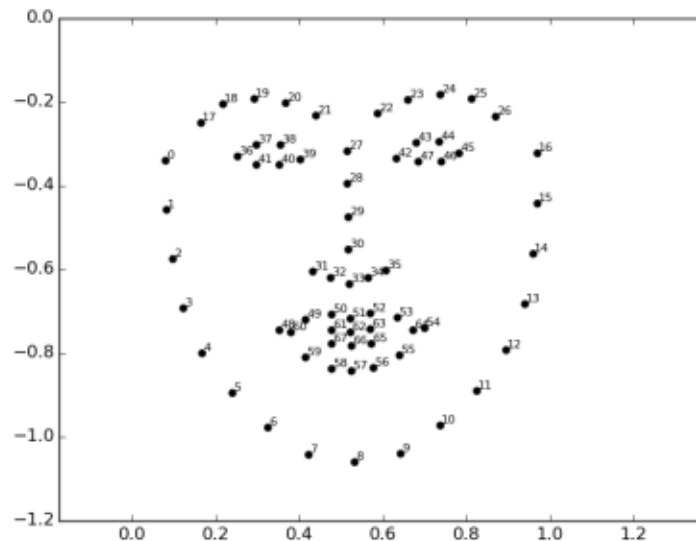


Рисунок 2.1 — Модель активної форми на 68 точок

Визначається даною формулою:

$$d(X, Y; S) = \sqrt{(X - Y)^T S^{-1} (X - Y)}, \quad (2.1)$$

де $X = (x_1 \dots x_N)^T$ – багатомірний вектор, від якого береться відстань;
 $Y = (y_1 \dots y_N)^T$ – середні значення множини, до якої рахується відстань;
 S – матриця коваріацій множини [17].

На основі схеми активної форми використовується триангуляція Делоне для множини пікселів на площині. Дана триангуляція, що жодний піксель множини не знаходиться всередині розписаних довкола трикутників кіл у цій сукупності. Таке розміщення дає максимально зменшити кількість незначних кутів трикутників.

2.3 Розробка архітектури нейромережі

За архітектурою нейромережі штучного виду поділяють на такі типи:

- багат шарові з прямою передачею даних;
- рекурентні;
- клітинні;
- багат шарові з оберненою передачею даних.

У багат шарових нейромережах з прямою передачею даних, до яких відноситься багат шаровий нейрони з'єднані в шари, між якими данні передається у одному напрямку. Таку структуру зображено на рисунку 2.2.

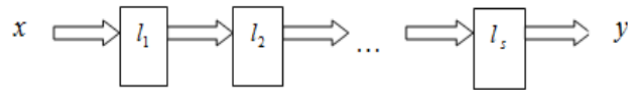


Рисунок 2.2 — Схема багат шарової нейромережі з прямим поширенням інформації

Результати обробки інформації з входу поступово переправляються по шарах мережі до кінця, де отримуємо остаточний вихідний нейромережевий імпульс. Загалом, необхідно зауважити, що кількість скритих шарів нейромережі може мати як завгодно багато. Це обумовлюється від складності поставленої місії.

Структура багат шарового персептрона наведено на малюнку 2.3., де трьома крапками вказано, що кількість скритих шарів і нейронів у них може бути безліч для кожної конкретної місії. Процедура тренування якраз полягає в визначенні даних параметрів і інколи є необхідно трудомістким процесом.

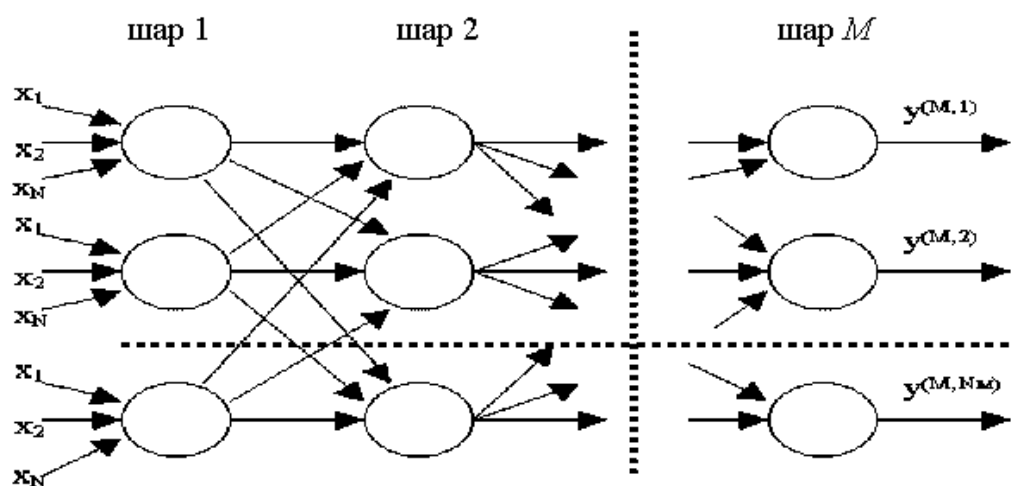


Рисунок 2.3 — Загальна архітектура багат шарового персептрона

В рекурентних неймережах штучного виду з виходу імпульси можуть подаватися на вхід або скриті шари, тобто дані мережі з зворотними зв'язками. Структура зазначеного типу неймереж проілюстровано на рисунку 2.4.

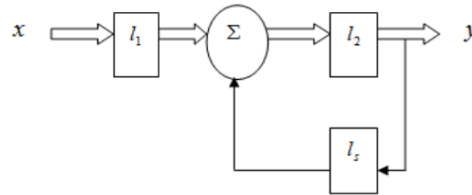


Рисунок 2.4 — Архітектура рекурентних нейронних мереж

У структурах клітинних неймереж кожен вузол зв'язаний з сусіднім вузлом. Дані структури мають значно складніші, оскільки дані від одного нейрона використовуючи синапатичні зв'язки передаються до всіх необхідних сусідніх нейронів. Зазначені структури потребують великих временних затрат на тренування.

Зважаючи на зазначенні недоліки, потік сигналів в системі може бути чи в одному напрямку чи повторюватися. В першому випадку йде розмова про динамічну архітектуру неймережі, оскільки вхідні імпульси подаються в вхідній шар, а потім після модифікації перенаправляються на наступний схований шар, як показано на малюнку 2.5.

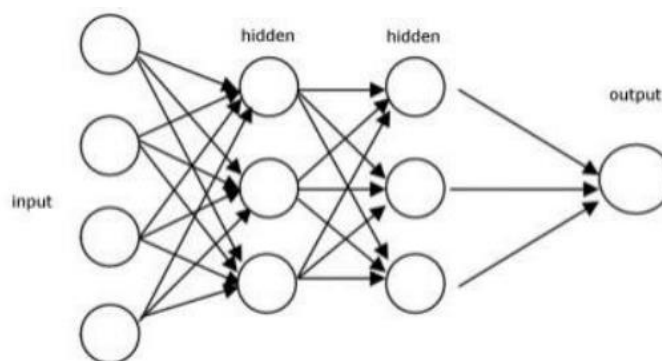


Рисунок 2.5 — Неймережа прямого поширення інформації

Якщо неймережа має визначений внутрішній повтор, то це символізує, що сигнали повертаються до вузла чи до шару, який вже отримував та

опрацьовував даний сигнал. В такому випадку нейромережа штучного виду має зворотний зв'язок, що продемонстровано на малюнку 2.6.

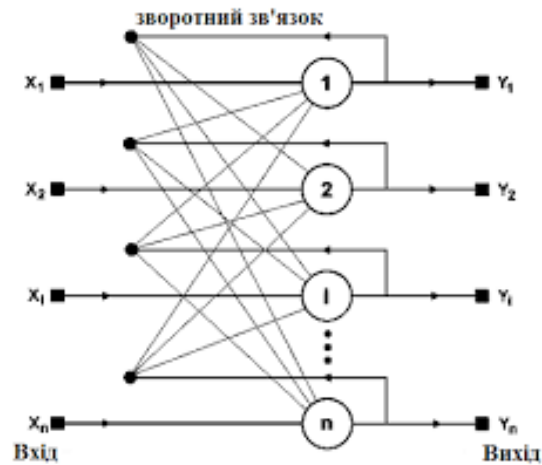


Рисунок 2.6 — Нейромережа зворотного поширення інформації

Головною причиною повторення у мережі є розроблення динамічної поведінки, спеціально коли вона розв'язує задачі, пов'язані із тимчасовими рядами чи ідентифікації образів, для яких необхідно внутрішня пам'ять, щоб підсилити процес навчання. Але такі нейромережі особливо важкі для тренування, а в кінцевому результаті регулярно не здатні тренуватися. Більшість нейромереж з зворотним зв'язком представлені одношаровими, зокрема, як мережі Ельмана та Хопфілда, однак можна побудувати та повторювану багатошарову мережу, для ілюстрації, багатошаровий перцептрон, але це вимагає суттєвих обчислювальних і часових ресурсів.

При застосуванні нейромережевих алгоритмів в задачах часто виникає потреба побудови нейромережевого шаблону об'єкта керування на основі отриманих вхідних і вихідних імпульсів в реальному часі. Застосуванні багатошарових перцептронів для розробки нейромережевої моделі є складним в зв'язку з тим, що екстрове навчання багатошарового перцептрону у певній ділянці робочої області зумовлює до втрати навченого стану в усій робочій зоні нейромережі, що не дає використовувати цей вид нейронних мереж для місій реального часу.

Зазначений мінус відсутній в радіально-базисних нейромережах (РБНМ), оскільки кожен їх елемент впливає на результат вихідного сигналу переважно лише в обмеженій ділянці робочої зони, який характеризується положенням центру об'єкту та параметром σ , що зветься шириною радіальної функції. Чим суттєвіше значення параметра σ , тим збільшується розмір зони, на яку впливає даний об'єкт.

Радіально-базисна нейромережа — ця структура, яка складається з трьох шарів: першого шару, на який подається вхідний імпульс-вектор, скритий шар, що складається із нейронів радіального виду і фінальний шар, який здійснює зважене підсумовування виводу роботи прихованого шару.

Скритий шар перетворює початковий вектор X , використовуючи різноманітні функції радіального фундаменту. Для створення ефективно функціонуючої системи використовують різні типи базисних функцій, зокрема, мультіквадратичну і функцію Гауса.

Алгоритми навчання таких нейромереж варіюють від алгоритму найменших квадратів до складних методів на основі градієнтів. Проте дані алгоритми вимагають попереднього місії кількості зв'язків в скритому шарі. Для налаштування ваг скритого шару, зазвичай, використовують алгоритми кластеризації.

Для поліпшення процедури моделювання вищеописаних структур нейромереж штучного виду застосовують різноманітні програмні методи, які детально розглянемо в наступному розділі.

Мережі CNN вперше були створені і використані приблизно у 80-х роках 20-го століття [17]. Найчастіше, на що вистачало ефективності ЕОМ тих часів для роботи із CNN — це ідентифікації рукописних цифр. Переважно вони використовувалася у поштовому секторі для сканування поштових індексів, пін-кодів і таке інше. Для тренування будь-якої моделі глибокого тренування потрібна велика кількість інформації, а також багато розрахованих ресурсів. Це було серйозним мінусом для CNN того періоду, та тому CNN скорочувались лише поштовими зоною, їм не вдалося отримати достатнього популярності.

У глибокому тренуванні згорткова нейронна мережа (CNN/ConvNet) — це тип глибоких нейронних мереж, які зазвичай застосовуються для дослідження візуальних малюнків [18]. Цей тип є спеціалізованим видом моделі нейронної мережі, необхідний для роботи із двовимірним набором, тобто даними рисунками, хоча їх можна використовувати із одновимірними і тривимірними даними. Коли мається на увазі синаптична мережа — зазвичай споминається про множення матриці, але це не є дійсності в ConvNet. Для CNN застосовується спеціальна методика під назвою згортка. Згортка в математиці — це математичний метод над двома функціями, яка в результаті створює третю функцію. Та дозволяє виражати як одна конфігурація змінюється іншою. Центральним в згортковій синаптичній мережі має згортковий шар, який та надав мережі назву. Саме даний шар виконує дію під назвою «згортка».

Шар згортки грає головну роль в CNN. Він складається із колекції математичних операцій спеціалізованого виду лінійної операції. В цифрових рисунках значення точок зберігаються у двовимірній (2D) сітці, тобто у наборі чисел, а невелика сітка ваг, що називається ядром (удосконалений екстрактор функцій) використовується до кожної позиції рисунку, що робить CNN ефективним для обробки рисунків. Оскільки один шар транслює свій вихід на наступний шар, розпізнанні об'єкти можуть ієрархічно і поступово виходить все більш складними. Мережева структури CNN натхнена функціонуванням зорової шару тварин. Дослідження поля зору виконується через використання набору субрегіонів, що розділяють фотографії. Кожен підрегіон досліджує нейрон для попередньої зміни невеликих обсягів даних. Це називається згортковою модифікацією. Структура згорткової нейронної мережі створюється послідовністю будівельних блоків для виокремлення ознак, які розрізняють належний тип образу від інших [19]. Будівельний фрагмент складається із однієї чи кількох частин.

Особливістю структурою мережі є те, як згорткові шари, зміна і об'єднання змінюють один одного у будівельних фрагмент, а також метод, яким самі будівельні фрагменти змінюють один одного.

Згорткове осередок має розмір фрагменту, часто 3 на 3 або 5 на 5. Аналізована зона (сприйнятливий поле) трохи більше, ніж осередок, оскільки додається етап, щоб сприйнятливі зони перекривалися. Цей метод дозволяє краще віддзеркалити зображення і покращити узгодженість його модифікації.

Процес оптимізації таких ваг, як ядра, називається тренування. Воно виконується таким чином, щоб зменшити різницю між виходами і основними мітками за допомогою методів оптимізації, який називається повторним поширенням та градієнтним спуском.

Дослідження характеристик зображення згортковим осередком є операцією фільтрації із важними показниками, пов'язаними із кожною точкою. Застосування фільтра до малюнку називається згорткою.

Далі згортки отримується карта характеристик, що є абстрактним представленням рисунку. Його значення залежать від характеристик застосованого ядра згортки і значень пікселів вхідного рисунку.

В контексті згорткової синаптичної мережі згортка — це прямолінійна операція, яка міститься множення набору характеристик коефіцієнтів на вхідні інформації, так само як це трапляється та в традиційних синаптичних мережах. Враховуючи, що методика була створена для двовимірних вхідних матеріалів, множення виконується між набором вхідних даних та двовимірним масивом параметрів, який називається фільтром чи ядром.

Кілька згорткових осередків проходять через рисунку, що призводить до кількох вихідних фрагментів функцій. Кожне згорткове осередок має дані, специфічні для даних, яка шукається в фото (наприклад: згортковий осередкового типу фільтра Собеля має характеристики для пошуку контурів на рисунку).

Фільтр менший за вхідну інформацію, а тип множення, який використовується між вхідним фрагментом розміром із фільтр, та фільтром є точковим добутком. Точковий добуток — це по елементне множення елементів розміру початкового фільтра на фільтр, яке потім розраховується, що призводить до об'єднання значення. Оскільки це призводить до остаточного значення,

операцію зазвичай називають «скалярним добутком» [20].

Застосування фільтра, меншого за початковий, є навмисним, оскільки дозволяє помножити один та той же фільтр (набір параметрів) на вхідний масив декілька разів в різних точках входу. Зокрема, фільтр використовується систематично до кожної частини, що наскладується, або фрагмента параметру фільтра вхідних даних, зліва на право, зверху до вниз [21].

Вибір характеристик згорткового ядра залежить від місії, яку необхідно розв'язувати. При алгоритмах глибокого навчання ці характеристики автоматично вивчаються алгоритмом з початкових даних. Зокрема, завдяки техніці градієнтного повторного поширення, яка дозволяє змінювати параметри відповідно до значення градієнта функції загуби. Функція втрат обчислює помилки між прогнозованим та цільовим значенням [22].

Систематичне використання одного й того самого фільтра на рисунку є фундаментальною концепцією. Якщо фільтр призначений для виокремлення певного типу об'єкта в вхідних даних, то систематичне використання цього фільтра по всьому початковому зображенню дає фільтру спроможність виявити цю функцію у будь-якому місці рисунку [23].

Витік одноразового множення фільтра на початковий масив є одне значення. Так як фільтр застосовується декілька разів до отриманого масиву, результатом є двовимірний набір вихідних значень, які відображають фільтрацію вхідних даних. Отже, двовимірний вихідний масив із цією маніпуляцією називається «картою ознак» [24]. Остаточне створення карти елементів можна передати кожний параметр в карті ознак через нелінійність, однаково, як це робиться для остаточних даних повністю з'єданого шару [25].

У цьому розділі було викладено послідовність обробки відео потоку, що пропонується у роботі для розпізнавання та ідентифікації особи. Також були наведені описи методів для окремих складових процесу розпізнавання. Детально розібрана й проаналізована згорткова нейронна мережа й умови її використання. Було доведено доцільність її застосування в магістерській роботі.

3 РОЗРОБКА ПРОГРАМИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ

У даному розділі розроблена структура програми розпізнавання людей у відеопотоці, сформована програма розпізнавання особи та виконано налаштування нейронної мережі.

3.1 Розробка структури програми розпізнавання людей

Для функціонування даної системи необхідні такі модулі, як отримання зображення, опрацювання зображення та розпізнавання особи (рисунок 3.1). Отримання зображення — відповідатиме за здатність системи отримати кадр із відеопотоку для розпізнавання. В собі може мати як інтерфейс для введення шляху до зображення, так і можливість працювати з камерами.

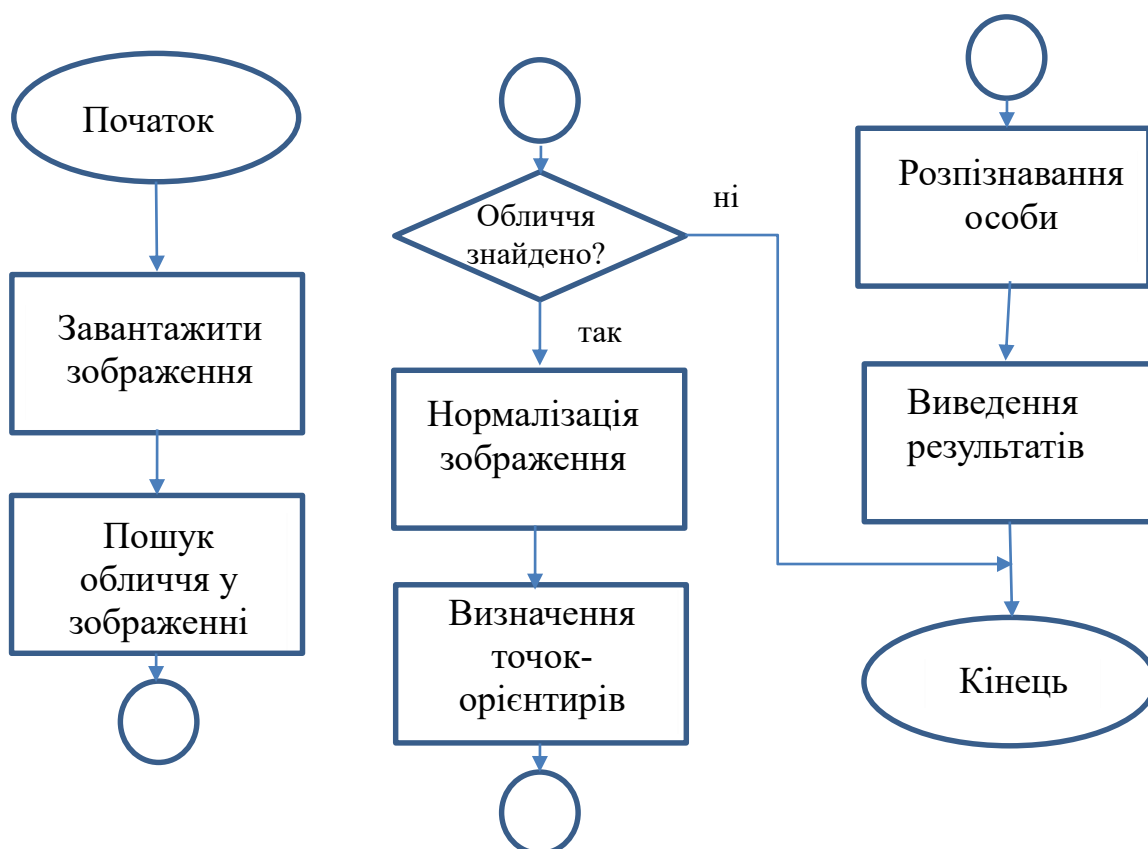


Рисунок 3.1 — Послідовність обробки зображення

Опрацювання кадру зображення — наступний елемент проекрованої системи, в якому використовуватимуться результати аналізу проведених вище. Основна місія це отримане зображення опрацювати та виділити в ній всі наявні лиця і відповісти рисунком з виділеними лицем. Діаграма використання модулів програми, що детально описує роботу програму із деталізацією по кожному прикладу, показано на рисунку 3.1.

Розпізнавання особи за зображенням обличчя — опісля система виокремлює унікальні точки орієнтирів лиць й проводить пошук подібних характеристик обличчя людини за допомогою SVM чи в наявній базі даних.

На діаграмі міститься три модуля, кожен із яких використовується лише після успішного закінчення попереднього модуля: від вхідного введення даних в систему до фінального — виведення результатів ідентифікації обличчя. Діаграма послідовності показує як передається фокус управління від одного модуля системи до інших.

Вказується шлях до файлу, у якому будемо розпізнавати особу (модуль Capture). Якщо файл не виявлений, то виводиться повідомлення про відсутність такого файлу, фактично про помилку. Якщо ж файл виявлений, то приступаємо до його опрацювання. Виділяємо із використанням детектора обличчя область, де є обличчя людини (модуль FaceDetector). Виділений фрагмент зображення нормалізуємо і проставляємо ключові точки, тобто точки-орієнтири на обличчі людини (модуль NormalizationFace). Підготовлений таким чином фрагмент зображення подаємо далі для розпізнавання на згорткову нейронну мережу (модуль RecognitionFace). Розпізнавання здійснюємо із залученням попередньо підготовлених даних (модуль FaceBase). Результат розпізнавання виводиться на екран монітору для користувача. Якщо у ході виконання процесу розпізнавання виділене обличчя людини було не розпізнано через його відсутність у базі даних, то виводиться повідомлення про факт відсутності цієї особи у базі даних. Для виконання операції розпізнавання згорткова нейронна мережа проходить етап попереднього навчання із використанням модуля LearningBlock. Розроблена

програма у вигляді UML-діаграми класів програмного засобу приведена у Додатку Д магістерської роботи.

Діаграма пакетів демонструє зв'язок між певними пакетами програми і визначає як вони діють між собою. Напрямую користувач взаємодіє з пакетом, який містить у собі елементи графічного інтерфейсу. Даний пакет взаємодіє із усіма іншими, передаючи їм дані для обробки зображення. Конектор забезпечує зв'язок із сервером, на якому зберігається сформована база даних. Це дозволяє виконувати усі завдання, що покладені на сформований програмний продукт.

3.2 Розробка програми розпізнавання людей

На даний момент існує багато архітектур нейронних мереж. Кожна відповідає для великої кількості місій, але між ними існує велика різниця в швидкості тренування, виконання і адаптації під різні системи, а також продуктивності на менш слабких технологіях. Тому програмісту потрібно вивчити всі деталі кожної структури, щоб вибрати найкращу для своєї місії. Серед такої великої кількості для ідентифікації образів була вибрана архітектура згорткової нейронної мережі. Після багаторазових аналізу та тестів серед дослідників і програмістів найкращі результати при рішенні задачі розпізнавання людей показала згорткова комп'ютерна нейрона мережа (англ. Convolutional Neural Network). Ця нейрона комп'ютерна мережа сприяла в майбутньому розробити такі структури як когнітрон і неокогнітрон. Вся успішність цієї архітектури завдання ідентифікації полягає в можливості враховувати двовимірну топологію рисунку, на відміну структури багат шарового перцептрона [30]. Ця архітектура нейронної мережі може неповністю забезпечити стійкість до різноманітного роду спотворення елементів, а саме: повороти, зміна кута, масштабу і поворотів. Згорткові нейронні комп'ютерні мережі містять в собі 3 архітектурні схеми, щоб дозволити ту саму адаптацію до просторових змін об'єкта:

— зменшення вагових характеристик за рахунок спільних вагів (також дає можливість знаходження схожих рис в будь-якому місці на рисунку);

- локальні рецепторні поля гарантують двомірну зв'язність в'язків;
- просторові підвибірки через ієрархічної організації нейроної мережі.

Щороку між дослідників проходить міжнародний співтовариство із розпізнавання об'єктів під назвою ImageNet. Між всіх структур згорткова НМ вважається однією із найефективнішою за точністю і швидкістю знаходження елементів на вхідних даних. З 2012 року нейроні комп'ютерні мережі почали посідати переможні місця в міжнародному конкурсі ImageNet. База інформації ImageNet — проект з створення і супроводу масивної бази інформації анованих рисунків, призначена для відпрацювання і тестування методів ідентифікації образів і машинного зору.

Елементи згорткової нейроної мережі. Згорткова нейрона комп'ютерна мережа складається із трьох типів шарів, які продемонстровані на рис. 3.2, та включає шар згортки, субдискретизуючий (підвибірковий) шар і перцептрон.



Рисунок 3.2 — Топологія згорткової нейроної мережі

Мова програмування Python має зручний функціонал та можливість використовувати новітньої технології, які прискорюють процес створення ПЗ. Так само ця мова забезпечує парадигму написання кросплатформових програм. Перед стартом роботи над проектом необхідно встановити інтерпретатор Python, після чого програміст може почати розробляти віртуальне середовища для роботи над системою. Це потрібно для того, щоб при розробці нового проекту встановлені елементи і фреймворки для одного проекту не іммігрували на інший, що може спричинити кучу похибок в процесі встановлення. Якщо застосовувати одне віртуальне середовище, то можна зіткнутися із проблемою, що на певний із

фреймворків може не вийти оновлення, в іто зі проблем з оновленням може виникнути похибки зворотної сумісності із іншими пакетами. Пропонується для кожного проекту розробляти нове віртуальне оточення. Віртуальне середовище дозволяє розробляти необмежену кількість віртуальних оточень на одній операційній системі.

Є декілька типів віртуального оточення для розробки на Python (virtualenv, ruenv, virtualenvwrapper). При розробці середовища для системи розпізнавання лица людини було вибрано інструмент virtualenv, який є одним з популярнішим виборів поміж розробників.

Для розробки програми, яка могла б створити і виконати тестування нейронної мережі застосуємо бібліотеку ImageIA — інструмент для ідентифікації різних об'єктів на динамічному рисунку чи фотографії на основі мови Python [31]. Для створення програми можна виокремити наступні етапи.

Для довантаження бібліотеки встановлюємо програму Python із пакетним менеджером pip для здобуття можливості встановлювати додаткові файли [32]. Розгортаємо IDE текстовий редактор PyCharm та в ньому створюємо новий проект з назвою image detection. Даний процес продемонстровано на рис. 3.3.

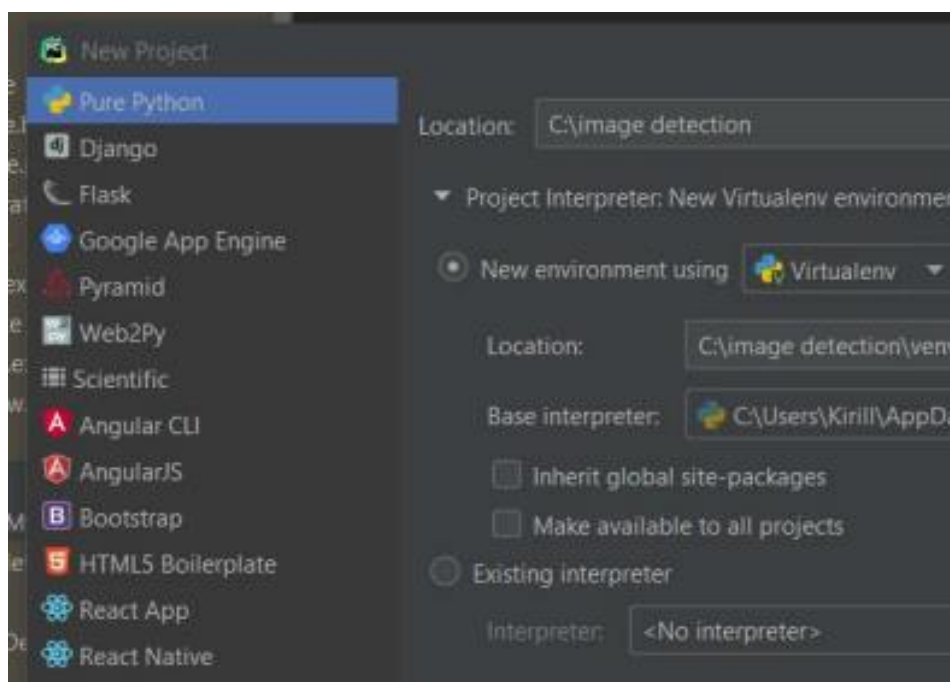


Рисунок 3.3 — Створення проекту

В результаті отримаємо робочу область. Використовуючи рядок `venv` створюємо новий файл — `Digodiy.py`.

Після розробки проекту і файлу отримуємо поле для створення коду програми із ідентифікації елементів зображення.

Далі перевіряємо додаток через тестовий запис «Dyplom Digodiy». У результаті використання цієї команди можна побачити, що відбулось підключення програми й бібліотек. Цей процес продемонстровано на рис. 3.4.

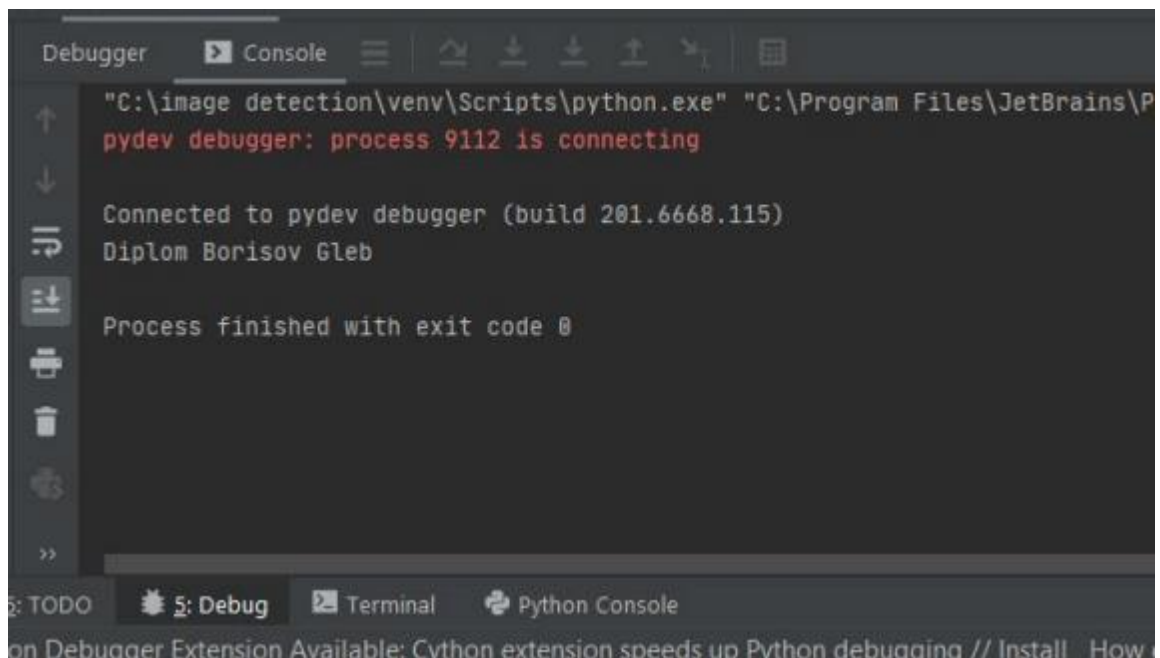


Рисунок 3.4 — Налаштування IDE

Наступним етапом є встановлення і завантаження необхідних файлів для забезпечення роботи й тестування комп'ютерних мереж. Зокрема, з'єднаємо файли Image AI: Object Detection. Для її роботи необхідно попередньо поставити додаткові модулі (бібліотеки). Дані модулі прописують у терміналі програми `pyCharm`.

Після цього етапу необхідно установити модель Yolo. Тоді як більшість програм використовують згорткову нейронну комп'ютерну мережу декілька разів з різними областями зображення, YOLO використовує її коли один раз до всього рисунку. Мережа ділить рисунки на сітку та передбачає розташування схожих об'єктів для кожної ділянки. YOLO працює значно швидше за R-CNN,

що є істотним показником для оброблення інформації у режимі реального часу на комп'ютерному пристрої. Має багато реалізацій в вигляді файлів, таких як Keras. Концепція розпізнавання об'єктів YOLO передбачає перехід до регресійну форму завдання. YOLO — одноетапний алгоритм глибокого навчання, який використовує згорткові нейронні мережі для виявлення об'єктів. На сьогоднішній день було розроблено кілька версій алгоритму. Принцип роботи YOLOv1 полягає в тому, що задане зображення поділяється на n -ну кількість однакових фрагментів, кожному з яких виявляється центр осередку. Кожен осередок передбачає фіксовану кількість обмежуючих рамок зі значенням достовірності (confidence score). В свою чергу, обмежувальні рамки базуються на п'яти складових: координатам по осі абсцис та ординат, висоті та ширині рамки, а також на значенні достовірності. Для вибору найбільш відповідного значення на наступному етапі алгоритм використовує IoU — перетин над об'єднанням. Крім того, YOLO включає опцію немаксимального придушення (NMS) для видалення зайвих рамок. Друга версія алгоритму доповнена пакетною нормалізацією (batch normalization) разом з згортковими шарами для зниження ймовірності перенавчання та підвищення точності. Відмітимо, що чисельність таких класів визначена на ступені 80 елементів.

Згідно з завданням на дослідження нас буде інтересувати саме тип об'єктів ідентифікації під номером 1. Окрім цього, наведений набір може свідчити про те, що синаптичні мережі можна застосовувати не лише для забезпечення дорожньої ситуації у програмах відеоспостереження, але й в побуті. Відтак, цю модель далі потрібно розмістити в ту ж папку, де розміщується файл Digodiy.py. Цим власне кінчається підготовчий етап до розробки програми із тестування нейронних мереж.

3.3 Навчання нейронної мережі

Для початку було вивчено два варіанта тренування згорткової нейронної схеми YOLOv5:

— перший — це тренувати модель із «нуля», тобто буде створено

параметри на основі структури із нульовими коефіцієнтами на нейронах;

— другий — використати попередньо треновану модель на наборі COCO.

Ці ваги створені і мають певні заключення про знаходження 80-ти класів елементів. Отже оцінки схеми за головними параметрами вказані на рис. 3.5.

teacher	student	mAP@.5	mAP@.5:.95	precision	recall
None	yolov5n	0.405	0.237	0.525	0.39
yolov5n	yolov5s	0.435	0.256	0.568	0.412

Рисунок 3.5 — Налаштування IDE

Тому маючи параметри з певним висновком було б не доцільно тренувати модель із повного «нуля». Було використано архітектуру YOLOv5s і обрано ваги, що були призначені на COCO.

Наступними етапом буде ітераційне навчання мережі із використанням створеного кастомного тренувального набору даних. Процес тренування буде складатись із поступового тренування та валідації кожного етапу. Необхідно розбити навчальний набір даних.

Для отримання етапного результату, було отримано три елементи тренувальних даних і одна перевірна. В абсолютних числах — це тренування у розмірі тисяча, дві і чотири тисячі рисунків і val — одна тисяча зображень. При запуску навчання моделі командою з рисунку 3.6 ми отримаємо такий результат до терміналу.

```
!python train.py --img 640 --batch 16 --epochs 100 --data /content
--cfg /content/train-yolov5/models/custom_yolov5s.yaml --weights .
```

Рисунок 3.6 — Команди для навчання моделі

На рисунку 3.7 можна побачити ініціалізацію параметрів архітектури YOLOV5s, де передається архітектура і ставиться відповідні гіперпараметри.

Після створення архітектури, встановлюються взаємини до навчальних і валідаційних даних. Також налаштовуються якорі для навчальної CNN.

Налаштувавши середовище, система запускає навчання моделі на 100 епох із розміром пакетів інформації у 16 фото та за параметрами 640 на 640 точок.

```

/content/yolov5
train: weights=../yolov5s.pt, cfg=/content/train-yolov5/models/custom_yolov5s.yaml, data=/content/tr
github: up to date with https://github.com/ultralytics/yolov5 ✓
YOLOv5 🚀 v6.2-198-gacff977 Python-3.7.15 torch-1.12.1+cud113 CUDA:0 (Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 🚀 in
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 🚀 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/Ultralytics/Arial.ttf...
100% 755k/755k [00:00<00:00, 121MB/s]

      from n   params  module
0         -1 1     3520  models.common.Focus          [3, 32, 3]
1         -1 1    18560  models.common.Conv           [32, 64, 3, 2]
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]
15        -1 1         0  torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
16       [-1, 4] 1         0  models.common.Concat         [1]
17        -1 1   95104  models.common.BottleneckCSP  [256, 128, 1, False]
18        -1 1  147712  models.common.Conv           [128, 128, 3, 2]
19       [-1, 14] 1         0  models.common.Concat         [1]
20        -1 1  313088  models.common.BottleneckCSP  [256, 256, 1, False]
21        -1 1  590336  models.common.Conv           [256, 256, 3, 2]
22       [-1, 10] 1         0  models.common.Concat         [1]
23        -1 1  1248768  models.common.BottleneckCSP  [512, 512, 1, False]
24       [17, 20, 23] 1  1634382  models.yolo.Detect           [601, [[10, 13, 16, 30,
custom_YOLOv5s summary: 233 layers, 8873294 parameters, 8873294 gradients

```

Рисунок 3.7— Ініціалізація мережі при тренуванні

Наступний фрагмент програми на рис. 3.8 відображає процес тренування мережі YOLOv5s на створеному наборі. На кожній ітерації можливо побачити висновки проміжних етапів навчання моделі. Зазвичай вони не занадто інформативні та направлені на представлення інформації про конкретний пакет початкових даних, що не дає підстав розробляти заключення про успішний тест або ні.

```

Transferred 145/369 items from ../yolov5s.pt
AMP: checks passed ✓
optimizer: SGD(lr=0.01) with parameter groups 59 weight(decay=0.0), 70 w
augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur
train: Scanning '/content/open-imgs-v6/train/labels' images and labels..
train: WARNING ⚠ /content/open-imgs-v6/train/images/000412e8f4509c60.jp
train: New cache created: /content/open-imgs-v6/train/labels.cache
val: Scanning '/content/open-imgs-v6/val/labels' images and labels...100
val: New cache created: /content/open-imgs-v6/val/labels.cache

AutoAnchor: 4.39 anchors/target, 0.997 Best Possible Recall (BPR). Curre

```

Рисунок 3.8 — Фрагмент програми по відображенню встановлення шляхів до навчальних даних й налаштування якорів

По закінченню навчання відображається узагальнена інформація про навчання, а саме вказується чисельність пройдених епох, час навчання та вказуються шляхи до отриманих ваг моделі. Є два результативних файли best.pt та last.pt. Метод адаптований для зберігання найкращої схеми у навчанні і кінцевої. На рисунку 3.9 відображається навчання і проходження навчання.

```

Plotting labels to runs/train/yolov5s_results/labels.jpg...
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/train/yolov5s_results
Starting training for 100 epochs...

```

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/99	5.53G	0.1041	0.1066	0.142	160	640: 100% 63/63 [01:05<00:00, 1.04it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 32/32 [00:22<00:00, 1.40it/s]
	all	1000	6909	0	0	0 0
1/99	8.25G	0.1001	0.1124	0.141	89	640: 100% 63/63 [01:01<00:00, 1.03it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 32/32 [00:23<00:00, 1.36it/s]
	all	1000	6909	0	0	0 0
2/99	8.25G	0.09805	0.1121	0.1394	96	640: 100% 63/63 [01:02<00:00, 1.00it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 32/32 [00:21<00:00, 1.48it/s]
	all	1000	6909	0	0	0 0
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
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
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
6/99	9.33G	0.08808	0.1156	0.1126	217	640: 100% 63/63 [01:00<00:00, 1.04it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 32/32 [00:25<00:00, 1.25it/s]
	all	1000	6909	0.000194	0.00655	0.00022 6.7e-05

Рисунок 3.9 — Відображення тренування та проходження навчальних епох

Після першої етапу тренування було проведено порівняння висновку, який був збережений в `./runs/train/yolov5s_results/weights/best.pt`, де є найкращий варіант навчених параметрів та попередніх, що були показані на рисунку 3.10.

```

Epoch      GPU_mem  box_loss  obj_loss  cls_loss  Instances  Size
99/99      9.33G   0.05925  0.09184  0.06805    107       640:
          Class   Images  Instances    P         R       mAP50
          all    1000    6909      0.556     0.0152    0.0103

100 epochs completed in 2.417 hours.
Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 18.1MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 18.1MB

```

Рисунок 3.10 — Результат навчання моделі

З результату порівняння можна зробити заключення, що тренування моделі на створених баз дає гарний приріст до ефективності моделі. Проте також необхідно переглянути характеристик на графіках для усіх тренувальних етапів, щоб переконатись у правильності створеного заключення. Порівняння `best.pt` і попередніх варіантів параметрів продемонстровано на рисунку 3.11.

Тому є підстави декларувати, що набір інформації сформований достатньо добре та дає можливість отримати необхідні результат на виході. Тому було реалізовано ще два тренування та валідація результуючих параметрів.

teacher	student	mAP@.5	mAP@.5:.95	precision	recall
None	yolov5n	0.405	0.237	0.525	0.39
yolov5n	yolov5s	0.435	0.256	0.568	0.412
yolov5s	yolov5_custom_1k	0.728	0.515	0.924	0.649

Рисунок 3.11 — Порівняння `best.pt` та попередніх варіантів ваг

Метрика програмного забезпечення (англ. *software metric*) — це міра, що

дозволяє отримати числове значення деяких властивостей програмного забезпечення та його специфікацій. Кількісні методи оцінювання добре показали себе в інших сферах науки, а тому багато теоретиків та практиків в галузі інформаційних технологій, спробували перенести цей підхід в розробку програмного забезпечення.

Узгодження — на цьому етапі тест-менеджер пояснює необхідність збору обраного набору метрик зацікавленим сторонам команди і замовнику відповідно. Оцінка — перевіряємо отримані дані. Обчислюємо значення показників на основі зібраних даних.

Після отримання результуючих параметрів згорткової нейронної мережі YOLOv5s, було здійснено тестування із «прогону» тестувальних даних через мережу.

Із перевірки проведеному на валідаційному наборі можна розробити заключення, що поставлену місію із навчання системи на створених тренувальних базах — виконано.

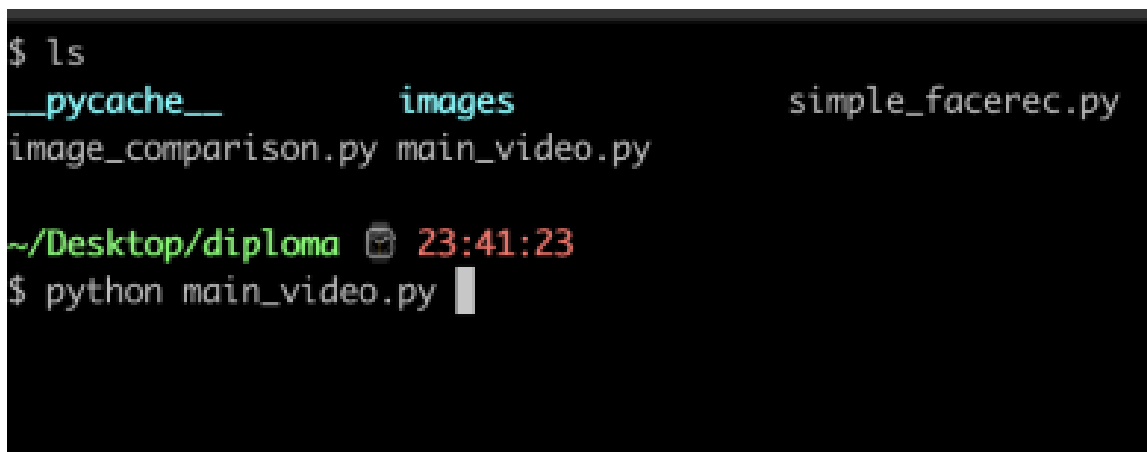
В цьому розділі була розроблена програма на мові Python та середовищі розробки PyCharm, також була взята архітектура згорткової нейронної мережі Yolo. Дана програма може розпізнавати обличчя в режимі реального часу із вебкамери чи відеокамери. Використовуючи її інтеграції в браузер є можливість далекого доступу і інші особливості, які гарантують зручність використання. І є функція ідентифікації певної особи по зображенню, що дає додаткові спроможності. Система непогано проявила себе в тестах з зусиллями ускладнити процес розпізнавання.

4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ

В даному розділі продемонстровані результати виконання застосунку для автоматизованої ідентифікації людини та виконана оцінка ефективності роботи програми

4.1 Перевірка якості роботи програми

Нижче розташовані рисунки із демонстрацією роботи програми. Для запуску проекту необхідно викликати певний файл “main_video.py” командою “python main_video.py”, знаходячись в корневій папці проекту, як показано на рисунку 4.1. На цьому ж рисунку проілюстровано виконання однієї із критерій до реалізації програми, а саме застосування Command Line Interface (CLI) як унікального способу запуску застосунку і демонстрації його результатів.



```
$ ls
__pycache__      images           simple_facerec.py
image_comparison.py main_video.py

~/Desktop/diploma 23:41:23
$ python main_video.py
```

Рисунок 4.1 — Графіки метрик навчання моделі

Після запуску програми одразу відкривається новітнє вікно, яке ілюструє в режимі реального часу результати дослідження.

Як можна побачити на зображенні рис. 4.2 обличчя автоматично виокремлено у червону рамку до якої підписано підпис з ім'ям. На даному ж рисунку продемонстровано як саме імітує реалізацію демонстрації результатів обробки розробленого застосунку для ідентифікації обличчя в режимі реального часу. В відкритому вікні одразу ілюструється весь відеоряд, отриманий із веб-камери і одразу додається червона рамка і підпис над нею, що гарантує змогу

майже моментально дати оцінку ефективності роботи програми і візуально порівняти очікування і фактичний результат відразу роботи застосунку.

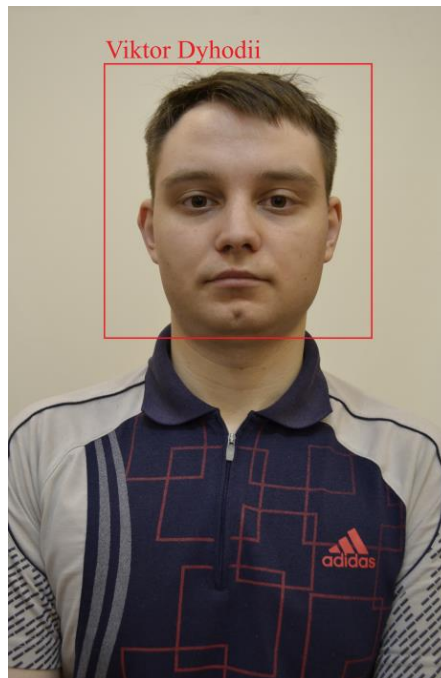


Рисунок 4.2 — Демонстрація результатів роботи застосунку

Інформація на зображенні, а саме ім'я, відповідає назві папки з тим самим обличчям в директорії “images”. Тобто найменування для рамки обирається самостійно, беручи за основу раніше збережені зображення. Таким чином, програма може не тільки в текстовому форматі демонструвати висновки розпізнавання лиця, а й візуально виокремити обличчя на відеоряді в режимі реального часу.

Значить, при використанні програми можливо одразу оцінювати продуктивність його роботи завдяки ілюстрації результатів в режимі реального часу. Реально, така демонстрація є лишньою при комерційному і цільовому використанню, адже самостійна система не має містити в собі фактичну ілюстрацією результатів роботи.

Створений застосунок може бути поліпшений шляхом розробки функціоналу фіксації нового обличчя через інтерфейс фотосесії на ту саму веб-камеру і збереження знімку в відповідній папці. Інші результати роботи програми продемонстровані на рисунках 4.3 та 4.4.

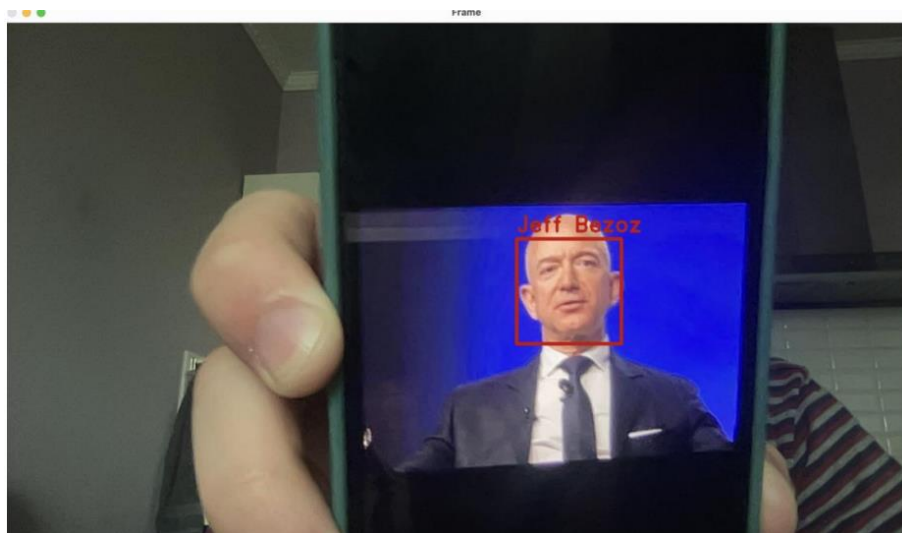


Рисунок 4.3 — Демонстрація роботи застосунку з іншим обличчям

Окрім того, задля отримання найбільшої ефективності є спроможність імплементувати більше алгоритмів попередньої обробки. Серед іншого, якщо додати в розроблений застосунок для ідентифікації обличчя метод аналізу кольору чи текстури шкіри, то можна покращити ефективність роботи програми до 20%. Проте, необхідно пам'ятати, що алгоритм аналізу текстури шкіру вимагає дуже високої роздільної спроможності камери, а це призводить до того, що тиск на систему зростає в багато разів.



Рисунок 4.4 — Демонстрація роботи застосунку з іншим обличчям

Так відбувається тому, що метод аналіз відео передбачає в собі те, що відео ділиться на багато фрагментів та кожен із них аналізується самостійно.

Крім того, необхідно пам'ятати, що при застосуванні методу дослідження текстури шкіри необхідно відмовитись від застосування методу зменшення рівня завад на зображенні, адже це може деградувати всю текстуру і метод аналізу шкіри просто не буде діяти. Також необхідно пам'ятати, що алгоритм бінарзації може негативно діяти на результати дослідження шкіри, тож необхідно понизити його інтенсивність.

4.2 Оцінка ефективності роботи програми

Спочатку буде приведено порівняння YOLO із іншими системами розпізнавання в реальному часі з використанням засобів PASCAL VOC. Щоб зрозуміти відмінності між варіантами YOLO та R-CNN, досліджено YOLO помилки VOC, розробленні YOLO і Fast R-CNN, однією із найбільш ефективних версій R-CNN. На основі різноманітних профілів похибок видно, що YOLO дозволяється використовувати для перегляду бистрих виявлень R-CNN і зменшення похибок з фонових помилкових переваг, що дає значне покращення продуктивності. Далі також проілюстровано результати VOC і порівняно середня точність mAP із новітніми методами. Нарешті, стає видно, що YOLO узагальнює новітні домени краще, ніж різні детектори на двох сетах даних для тестування рисунку.

Модель R-CNN поєднує метод вибіркового пошуку для виявлення пропозицій регіонів і глибоке навчання для виявлення об'єкта в цих регіонах. Розмір кожної пропозиції регіону змінюється, щоб відповідати вхідним даним CNN, з яких витягуємо вектор ознак із 4096 вимірами. Вектор ознак передається до декількох класифікаторів для отримання ймовірностей приналежності до кожного класу. Кожен із цих класів має класифікатор SVM, навчений робити висновки про можливість виявлення цього об'єкта для заданого вектору ознак. Кожен із цих класів має класифікатор SVM, навчений робити висновки про можливість виявлення цього об'єкта для заданого вектору ознак.

Багато дослідницьких зусиль із розпізнавання об'єктів зосереджуються на швидкому підвищенні стандартних процесів виявлення. Однак, лише Sadeghi та інші насправді пропонують систему виявлення, яка працює в режимі реального часу (30 кадрів в секунду або краще). Порівнюється YOLO з їх GPU-реалізацією DPM, яка діє чи на 30 Гц, чи на 100 Гц. Хоча різні метод не є настільки швидкими, щоб передбачати результати у режимі реального часу, в цьому місці все одно буде наведено компарація цих систем із метою аналізу компромісу між точністю схеми і її швидкістю у задачі виявлення об'єктів.

Було застосовано тренування YOLO, застосовуючи модель VGG-16. Дана модель є більш строгою, але також набагато повільнішою, ніж YOLO. Це ефективно для порівняння із різними системами розпізнавання, які покладаються на VGG-16, але отже він проходить повільніше, ніж в режимі реального часу, решта аналізів зосереджується на швидких схемах.

R-CNN мінус R міняє селективний пошук на пропозиції з статичними стримувальними рамками. Хоча це набагато ефективніше, ніж R-CNN, йому усе ще не вистачає швидкості реального часу та значно губить точність, оскільки в нього немає непоганих пропозицій. Faster R-CNN пришвидшує етап класифікації R-CNN, але його все ще покладається на відбірковий пошук, який може зайняти коло 2 секунд на рисунку для створення пропозицій обмежувальних прямокутників.

Було розроблено порівняння властивості YOLO із Fast R-CNN на наборі VOC, отже Fast R-CNN має один із найвищих параметрів детекторів на PASCAL та його виявлення є публічними. Для цього було застосовано методологію і інструменти Noiem et al. Для кожної категорії у тестовий час аналізуються N основних прогнозів для даної категорії. Кожне передбачення чи правильне, чи його класифікують за видом помилки:

- правильно: правильний клас та $IOU > 0.5$;
- локалізація: правильний клас, $0.1 < IOU < 0.5$;
- подібні: клас схожий, $IOU > 0.1$;
- інше: клас неправильний, $IOU > 0.1$;

— фон: $IOU < 0.1$ для будь-якого класу.

YOLO старається правильно локалізувати елементи. Помилки локалізації виділяються на більше помилок Fast R-CNN, ніж всі інші джерела загалом. YOLO робить набагато менше похибок локалізації, і значно менше фонових похибок. 13,6% найкращих призначень — це помилкові плюси, які не містять жодних елементів. YOLO майже у 3 рази частіше прогнозує заднє виявлення, ніж Fast R-CNN.

YOLO виконує набагато менше задніх помилок, ніж Fast R-CNN. Застосовуючи YOLO для ліквідація фонових виявлень від Fast R-CNN, отримують значне покращення продуктивності. Для кожного обмежувальної області, яке прогнозує R-CNN, перевіряємо, чи YOLO прогнозує подібне поле. Якщо це так, надається цьому аналізу майбутнього підсилення, засноване на вірогідність, передбаченій YOLO, і перекритті між двома областями.

Комбінована схем Fast R-CNN + YOLO — один із найефективніших алгоритмів виявлення. Fast R-CNN набуває на 2,3% удосконалення від комбінації із YOLO. Покращення від YOLO — це не просто супутній продукт зливання схем, оскільки від об'єднанні різних версій Fast R-CNN мало ефективність. Скоріше, саме тому, що YOLO виконує різні типи помилок в тестовий час, вона настільки швидка для підвищення продуктивності Fast R-CNN.

На жаль, дана комбінація не має переваги від швидкості YOLO, враховуючи кожна модель запускається самостійно, а потім поєднуються висновки. Однак, YOLO настільки швидкий метод, він не додає суттєвого часу на розрахунок порівняно із Fast R-CNN.

Для інформації, середня точність (Average Precision — AP) розраховується на наборі даних VOC по типу «особа», де усі моделі тренуються лише за інформацією VOC. На Picasso моделі навчаються на VOC, тоді як на People-Art вони навчаються на VOC.

R-CNN має високий рівень AP на VOC. Але R-CNN значно програє, коли використовується до художнього твору. R-CNN застосовується селективний пошук для варіантів із обмежувальним вікном, настроєним на природні рисунки.

Класифікаційний етап в R-CNN бачить лише дрібні регіони і потребує гарних варіантів.

YOLO має хороші показники на VOC, і його AP деградує менше, ніж інші методи, коли застосовується. Як та DPM, YOLO моделює розміри і форму об'єктів, а також взаємини між об'єктами і фрагментами, де об'єкти зазвичай появляються. Художнє фото і природні зображення дуже розходяться на піксельному рівні, але схожі за різними параметрами і формою предметів, тому YOLO все ще може прогнозувати хороші обмежувальні коробки і самі виявлення.

При тестування розробленої програми її було порівняно з новітньою системою на OpenCV при різних факторах. Результати дослідження представленні в таблиці 4.1.

Таблиця 4.1 — порівняння YOLO з іншими системами виявлення об'єктів

Назва системи	Недалеко	Далеко	Обличчя знаходиться під кутом	Погана яскравість
Система на OpenCV	правильно	Не завжди правильно	Не завжди правильно	Не завжди правильно
Розроблена система	правильно	правильно	правильно	правильно

Система на OpenCV чудово впоралася із близькою дистанцією, у інших експериментах фіксувалися проблеми із розпізнаванням конкретної персони. Розроблена система впоралась чудово у всіх експериментах.

На наступних ілюстраціях показано критерії виявлення (число вірно розпізнаних рисунків з загальної вибірки, час ідентифікації, пам'ять, яка використовується) і експерименти, які були при аналізі систем. На рис. 4.5 представлено результати використаної пам'яті на основі системи на OpenCV.

OpenCV — це бібліотека функцій програмування, головним чином для комп'ютерного зору в реальному часі. Спочатку розроблений компанією Intel, пізніше він був підтриманий Willow Garage, потім Itseez (який пізніше був

придбаний Intel). Бібліотека є кросплатформною та ліцензована як безкоштовне програмне забезпечення з відкритим вихідним кодом за ліцензією Apache License2.

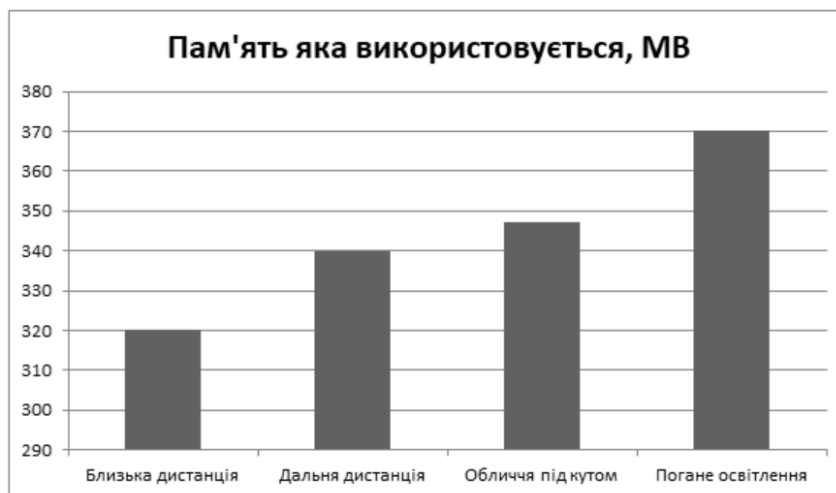


Рисунок 4.5 — Система на OpenCV (пам'ять, що використовується)

На рис. 4.6 представлено результати часу розпізнавання на основі використання системи на OpenCV.



Рисунок 4.6 — Система на OpenCV (час розпізнавання)

На рис. 4.7 представлено результати використаної пам'яті для здійснення операції розпізнавання на основі розробленої системи.

Проблема оцінки якості створених програмних засобів завжди була і є дуже важливою задачею, рішення якої дозволяє на ранніх стадіях життєвого циклу визначити характеристики програми, яка створюється. Існує ряд методик,

які направлені на оцінку параметрів майбутніх програм, що дозволяє адекватно оцінити трудовитрати і вартість розробки. Серед них представлено метрику Холстеда, яка оцінює характеристики програмних засобів на основі лексичного аналізу тексту програм.



Рисунок 4.7 — Розроблена система (пам'ять, що використовується)

На рис. 4.8 представлено результати по затраченому часу розпізнавання на основі розробленої системи.

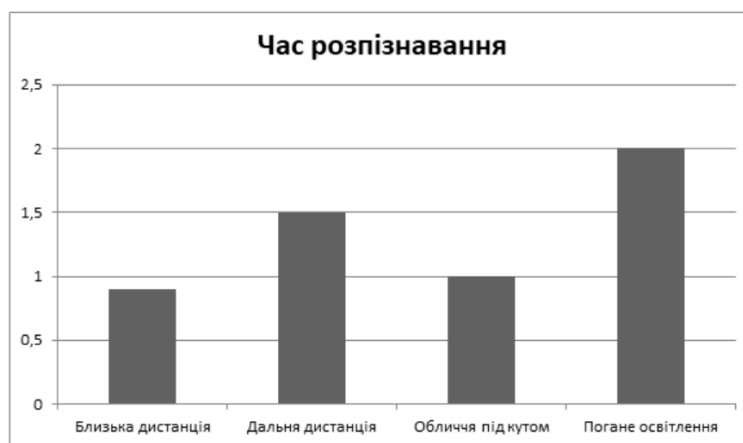


Рисунок 4.8 — Розроблена система (пам'ять, яка використовується)

По даних, які показані на графіках, необхідно сказати що система на OpenCV вимагає більший об'єм оперативної пам'яті (320 Мб проти 80 мб), а також потребує більших затрат часу на здійснення операції розпізнавання.

Узагальнюючи результати експериментів можна відзначити, що розроблена система на базисі нейронних мережах демонструвала себе краще ніж

система на базисі OpenCV. У експерименті на близькій відстані обидві системи впорались із розпізнаванням. Проте у інших експериментах система на базисі OpenCV мала індивідуальні проблеми із ідентифікацією.

Обсяг вжитої пам'яті був більший на системі на базисі OpenCV (приблизно 300 Мб проти у середньому 80 Мб на створеній системі), а час ідентифікації був також більший на даній системі. Також необхідно виділити, що в системі OpenCV фіксувались значні проблеми із FPS (кількість кадрів у секунду), коли на системі із нейронними мережами даної проблеми не було.

Якщо зазначати про переваги системи на базисі OpenCV, то необхідно відзначити, що таку систему легше розробляти через простоту використання алгоритмів із цієї бібліотеки комп'ютерного зору.

В даному розділі було виконана перевірка якості розробленої системи та відзначено її переваги порівняно із іншими системи розпізнання людей.

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ

Метою економічної частини магістерської кваліфікаційної роботи є довести економічну доцільність та ефективність впровадження наукової розробки, для цього необхідно виконати такі етапи:

- оцінити комерційний потенціал розробки;
- спрогнозувати витрати на виконання наукової роботи та впровадження її результатів;
- спрогнозувати комерційний ефект від реалізації результатів розробки;
- розрахувати ефективність вкладених інвестицій та період їх окупності.

5.1 Оцінювання економічного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Для проведення технологічного аудиту залучено трьох незалежних експертів. Кожен з експертів повинен ознайомитися з запропонованою розробкою, та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Для проведення технологічного аудиту залучено трьох незалежних експертів. Кожен з експертів повинен ознайомитися з запропонованою розробкою, та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Оцінювання комерційного потенціалу розробки здійснюється за критеріями, наведеними в таблиці 5.1 [33].

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

Критерії оцінювання та 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
Ринкові переваги (недоліки):					
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
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 динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Кінець таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
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 і з 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
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

Результати оцінювання комерційного потенціалу наведено в табл. 5.2.

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	4	3	3
2	3	2	2
3	4	3	3
4	4	2	2
5	3	2	3
6	2	2	1
7	2	1	1
8	3	2	2
9	1	1	1
10	4	4	4
11	4	3	3
12	4	4	4
Сума балів	СБ ₁ =38	СБ ₂ =29	СБ ₃ =29
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^i СБ_i}{i} = \frac{38 + 29 + 29}{3} = 32$		

Отже, з отриманих даних таблиці 5.2 видно, що середньоарифметична сума балів дорівнює 32, тобто нова розробка має рівень комерційного потенціалу вище середнього.

5.2 Розрахунок кошторису витрат на розробку програм

Прогнозування витрат на виконання науково-дослідної чи дослідно-конструкторської роботи може складатися з таких етапів:

- розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи.
- розрахунок загальних витрат на виконання даної роботи.
- прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Розрахунок витрат можна здійснити за такими статтями та формулами:

Основна заробітна плата із розробників (дослідників) Z_o , якщо вони працюють в наукових установах бюджетної сфери [33]:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн.}, \quad (5.1)$$

де M — місячний посадовий оклад конкретного розробника, наукового керівника;

T_p — кількість робочих днів у місяці, $T_p = 23$ дні.

t — число робочих днів роботи розробника (дослідника).

Заробітна плата розробника:

$$Z_p = \frac{11000}{23} \cdot 65 = 31086,95 \text{ (грн)}$$

Заробітна плата наукового керівника проекту:

$$Z_{HK} = \frac{9000}{23} \cdot 22 = 8608,69 \text{ (грн)}$$

Витрати на оплату праці, основна заробітна плата:

$$Z_o = Z_p + Z_{HK} = 31086,95 + 8608,69 = 39695,64 \text{ (грн)}$$

Таблиця 4.3 — Розрахунки основної заробітної плати спеціалістів

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Розробник	11000	478,26	65	31086,95
Науковий керівник проекту	9000	391,3	22	8608,69
Всього:				39696,64

Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даної роботи, розраховується як $(10...12)\%$ від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = (0,1...0,12) \cdot Z_o \quad (5.2)$$

Додаткова заробітна плата усіх робітників та розробників:

$$Z_d = 0,12 \cdot 39696,64 = 4763,59 \text{ (грн)}$$

Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даної роботи, розраховуються за формулою:

$$H_{зп} = (Z_o + Z_p + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де Z_o — основна заробітна плата розробників, грн.;

Z_d — додаткова заробітна плата розробників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %.

З 01.01.2018 року ставка єдиного внеску на загальнообов'язкове державне соціальне страхування встановлено 22%.

Нарахування на заробітну плату:

$$H_{зп} = (39696,64 + 4763,59) \cdot \frac{22}{100} = 9781,25 \text{ (грн)}$$

Амортизаційні відрахування обладнання можна розрахувати за формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн.}, \quad (5.4)$$

де Ц — загальна балансова вартість обладнання;

H_a — річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = 10-25\%$;

T — термін використання обладнання, T = 2-4 міс.

Витрати на матеріали M, що були використанні під час виконання роботи, розраховуються по кожному виду матеріалів за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_B \text{ грн.}, \quad (5.5)$$

де H_i — витрати матеріалу i -го найменування;

$Ц_i$ — вартість матеріалу i -го найменування, грн./кг.;

K_i — коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

B_i — маса відходів матеріалу i -го найменування, кг;

$Ц_B$ — ціна відходів матеріалу i -го найменування, грн/кг.

Таблиця 5.4 — Результати розрахунків амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1.Ноутбук	10000	15	3	375
2. Приміщення лабораторії	58000	15	3	2175
3. Камера.	350	25	4	29
4. Програмне забезпечення.	4400	20	2	147,4
5. Програма для розпізнавання обличчя.	9819	10	3	245,48
Всього				A = 2800

Таблиця 5.5 — Вартість матеріалів, що були використані для розробки ПЗ

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Папір	Шт.	100	1	100
Всього				100

Загальна вартість витрат матеріалів становить:

$$M = 100 \cdot 1,1 = 110 \text{ (грн)}$$

Використані для розробки нового технічного рішення комплектуючі наведено в табл. 5.6.

Витрати на комплектуючі вироби (B_K) розраховуються згідно з їх номенклатурою за формулою:

$$B_K = \sum_i H_{Ki} \cdot C_{Ki} \cdot K, \quad (5.6)$$

де H_{Ki} — кількість комплектуючих i -го найменування;

C_{Ki} — ціна комплектуючих i -го найменування;

K — коефіцієнт транспортних витрат.

Транспортні витрати на підприємствах складаються з постійних та змінних витрат, що беруть участь в процесі транспортування.

Таблиця 5.6 — Комплектуючі, що використовуються для розробки нового технічного рішення

Найменування комплектуючих	Кількість, шт.	Ціна за одиницю, грн.	Сума, грн.
1.Мікропроцесорна платформа. (350 грн)	1	350	350
2,Відеокамера (1092 грн).	1	1092	1092
3.Лазер (650 грн).	1	650	650
4,Кроковий двигун (399 грн).	1	399	399
5.Електронні компоненти (200 грн).	1	200	200

Всього, грн.			2691

Витрати на силову електроенергію V_e , розраховується за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \text{ грн.}, \quad (5.7)$$

- де V — вартість 1кВт-год. електроенергії, $V = 6,7$ грн/кВт [34];
 Π — установлена потужність обладнання, $\Pi = 0,06$ кВт;
 Φ — фактична кількість годин роботи обладнання, $\Phi = 455$ год;
 K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,75$.
Витрати на силову енергію становлять:

$$V_e = 6,7 \cdot 0,06 \cdot 455 \cdot 0,75 = 137,18(\text{грн})$$

Інші витрати I_v можна прийняти як (100...300%) від суми основної заробітної плати розробників, тобто:

$$V_{in} = (1...3) \cdot 3_0 \quad (5.8)$$

Інші витрати становлять:

$$V_{in} = 2 \cdot 39696,64 = 79393,28(\text{грн})$$

Сума усіх витрат становить:

$$V_{in} = 39696,64 + 4763,59 + 9781,25 + 2800 + 137,18 + 2691 + 110 \\ + 79393,28 = 139\,372,94(\text{грн})$$

Розрахунок загальних витрат на виконання роботи. Загальна вартість роботи визначається за $V_{\text{заг}}$ формулою:

$$B_{\text{заг}} = \frac{B}{\alpha} \quad (5.9)$$

де α — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях $\alpha = 1$.

$$B_{\text{заг}} = \frac{139\,372,94}{1} = 139\,372,94 \text{ (грн)}$$

Прогнозування загальних витрат на виконання та впровадження результатів. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta}, \quad (5.10)$$

де β — коефіцієнт, який характеризує етап (стадію) виконання даної роботи, на стадії розробки промислового зразка, $\beta \approx 0,7$.

$$ЗВ = \frac{139\,372,94}{0,7} = 199\,104,2 \text{ (грн)}$$

5.3 Розрахунок експлуатаційних витрат для нової розробки

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Позитивні результати від впровадження розробки очікуються протягом трьох років від впровадження розробки. Одним із основних позитивних результатів є зростання величини прибутку. Зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i, \quad (5.11)$$

де $\Delta\Pi_{\text{я}}$ — покращення основного якісного показника від впровадження результатів розробки у даному році;

N — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати матеріалів на розробку алгоритму зменшаться на 250 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 250 грн), а кількість користувачів збільшиться: протягом першого року — на 200 користувачів, протягом другого року — на 150 користувачів, протягом третього року — на 100 користувачів.

Реалізація продукції до впровадження результатів наукової розробки складала 600 шт., а прибуток, що його отримувало підприємство (організація) на одиницю продукції до впровадження результатів наукової розробки — 400 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого продукту $\Delta\Pi_i$ протягом першого року складатиме:

$$\Delta\Pi_1 = 250 \cdot 600 + (400 + 250) \cdot 200 = 280000 \text{ (грн)}$$

Протягом другого року:

$$\Delta\Pi_2 = 250 \cdot 600 + (400 + 250) \cdot (200 + 150) = 377500 \text{ (грн)}$$

Протягом третього року:

$$\Delta\Pi_3 = 250 \cdot 600 + (400 + 250) \cdot (200 + 150 + 100) = 442500 \text{ (грн)}$$

Чистий прибуток це прибуток після сплати податків. Прибуток також визначають як суму, на яку зріс власний капітал компанії за даний період у результаті діяльності цієї компанії.

5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

В даному підрозділі необхідно кількісно спрогнозувати, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної роботи.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

Крок 1 — розрахуємо теперішню вартість інвестицій PV , що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат $ЗВ$ на виконання та впровадження результатів НДДКР, розраховану раніше, тобто будемо вважати, що $ЗВ = PV = 199104,2$ (грн).

Крок 2 — крок. Розрахуємо очікуване збільшення прибутку $\Delta\Pi_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta\Pi_1 = 280000 \text{ (грн)}, \quad \Delta\Pi_2 = 377500 \text{ (грн)}, \quad \Delta\Pi_3 = 442500 \text{ (грн)}.$$

Крок 3 — для спрощення подальших розрахунків необхідно побудувати вісь часу, на яку наносять всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Якщо загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнюють 199104,2 грн., а результати вкладених у наукову розробку інвестицій почнуть виявлятися вже вкінці другого року впровадження. То ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 280000 грн. відносно базового року, у другому році — збільшення чистого прибутку на 377500 грн. (відносно базового року), у третьому році — збільшення чистого прибутку на 442500 грн. (відносно базового року).

Тоді рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 5.1.

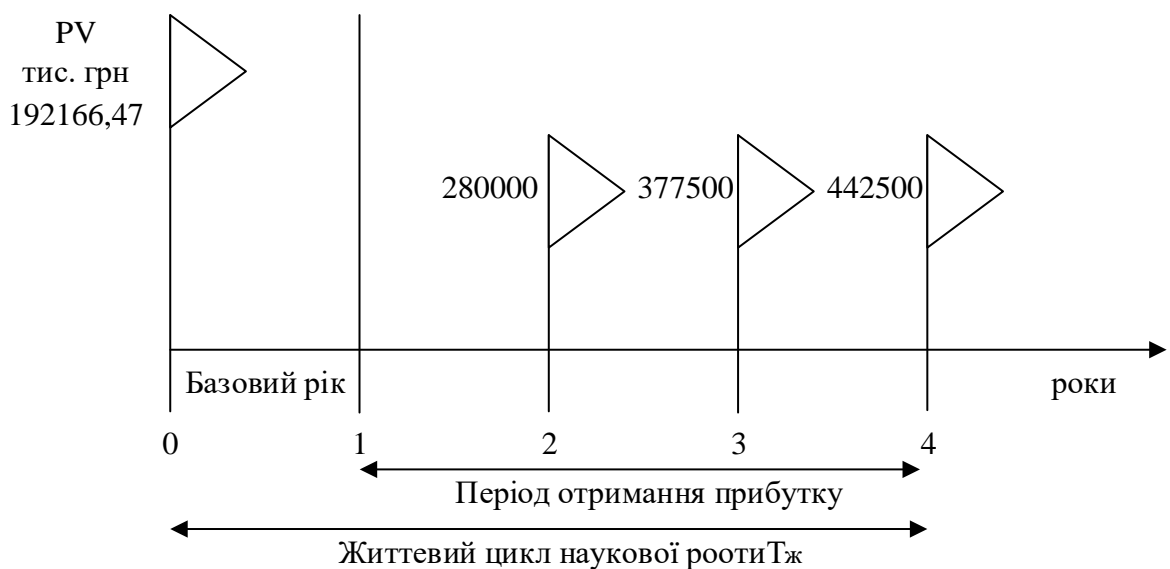


Рисунок 5.1 — Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Крок 4 — розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$. Якщо $E_{\text{абс}} \leq 0$ то результати від проведення наукових досліджень та їх впровадження буде збитковим і вкладати гроші в проведення цих досліджень ніхто не буде.

Для цього показника скористаємося формулою:

$$E_{abc} = (\text{ПП} - \text{PV}), \quad (5.12)$$

де ПП — приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій $PV = 3B = 199104,2$ грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.13)$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t — період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$\text{ПП} = \frac{280000}{(1 + 0,1)^2} + \frac{377500}{(1 + 0,1)^3} + \frac{442500}{(1 + 0,1)^4} = 231404,95 + 283621,33 + 302233,45 = 817259,73 \text{ (грн)}$$

$$E_{abc} = (817259,73 - 199\ 104,2) = 618\ 155,53$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Крок 5 — розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v . Для цього використаємо формулу:

$$E_v = T_{\text{ж}} \sqrt{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.14)$$

де $E_{\text{абс}}$ — абсолютна ефективність вкладених інвестицій, грн;

PV — теперішня вартість інвестицій $PV = 3B$, грн;

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

Далі, розрахована величина E_v порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f \quad (5.15)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = (0,14...0,2)$;

f — показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,1)$, але може бути і значно більше.

Якщо величина $E_v > \tau_{\text{мін}}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде.

Спочатку спрогнозуємо величину $\tau_{\text{мін}}$. Припустимо, що за даних умов $\tau_{\text{мін}} = 0,18 + 0,07 = 0,25$.

Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_B = \sqrt[4]{1 + \frac{618\,155,53}{199104,2}} - 1 = 0,42$$

Оскільки $E_B = 42\% > \tau_{\text{мін}} = 25\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Крок 6 — розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ОК}}$ можна розрахувати за формулою:

$$T_{\text{ОК}} = \frac{1}{E_B} \quad (5.16)$$

Якщо $T_{\text{ОК}} < 3\dots 5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним. В інших випадках потрібні додаткові розрахунки та обґрунтування.

Термін окупності становить:

$$T_{\text{ОК}} = \frac{1}{0,42} = 2,38 \quad (5.17)$$

$T_{\text{ОК}} < 5$ років, що свідчить про доцільність фінансування даної наукової розробки.

Рівень комерційного потенціалу нової розробки є вище середнього. Показники ефективності показують, що даний метод є доцільним і буде цікавий для інвестора. Термін окупності розробленого проекту менше 5-ти років, що підтверджує доцільність вкладання коштів в дану розробку. Економічний ефект від впровадження розробленої програми очікується позитивним при невеликій вартості і складності розробки даного ПЗ, це зробить розпізнавання обличчя у різних ситуаціях більш доступним у галузі кібербезпеки та охоронній діяльності.

ВИСНОВКИ

В рамках кваліфікаційної магістерської роботи був описаний та реалізований програмний засіб для розпізнання обличчя людей у відео потоці.

У роботі було виконано усі поставлені задачі, а саме:

— проаналізовано методи визначення головних точок у отриманих зображеннях, детально розглянуто поняття головних точок й кутів, забезпечено огляд найбільш відомих алгоритмів знаходження ключових точок обличчя, визначено їх плюси та мінуси, розглянуто проблеми, що зв'язані із визначенням та розміщенням ключових точок при розпізнаванні обличчя;

— виконано дослідження алгоритмів класифікації зображень на підставі їх статистичних особливостей, розглянуто найбільш поширені класифікатори, вказано їх переваги й недоліки, а також проаналізовано класифікатор, що базується на аналізі ймовірнісних властивостей зображення;

— досліджено наявні двовимірні методи до ідентифікації обличчя людей, визначено чотири базових підходи до пошуку та виокремлення обличчя та надано загальний опис тих алгоритмів, що відносяться до кожного із описаних підходів розпізнавання;

— створено методику ідентифікації обличчя людини на отриманих зображеннях із відео потоку, в основі яких знаходиться застосування моделі активної форми у вигляді ключових точок та форми контуру обличчя людини для її розпізнавання, визначено кроки функціонування програми та обрано оптимальні підходи до виконання кожного із цих кроків;

— сформовано перелік обов'язкових інструментальних засобів для розробки програмного продукту;

— виконано усі етапи створення програмного засобу для розпізнавання обличчя людини на отриманих кадрах відео потоку, де на всіх етапах роботи здійснено різноманітні програмні й алгоритмічні оптимізації, що дозволили втілити досить гнучку послідовність роботи програми та її взаємодії із користувачем;

— проведено тестування створеного програмного застосунку, за результатами якого були визначені особливості роботи даного підходу, умови його використання та визначення його продуктивності, яка для даного програмного застосунку є достатньою;

— виконані економічні розрахунки із обґрунтування доцільності виконання нової роботи по розробці нейромережевої технології виявлення і розпізнавання людей у відеопотоці, обчислені фінансові затрати на створення програмного продукту та визначено економічні переваги від впровадження запропонованого наукового рішення у вигляді завершеного програмного продукту.

Створений програмний продукт може використовуватися у комп'ютерних системах відеоспостереження для виявлення і розпізнавання людей у відеопотоці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zhang X. Real-Time Face Detection and Recognition in Complex Background. / X. Zhang, T. Gonnot, J. Saniie, // Journal of Signal and Information Processing, 2017, 8. — Pp. 99 — 112. doi: [10.4236/jsip.2017.82007](https://doi.org/10.4236/jsip.2017.82007).
2. Li L. Review of Face Recognition Technology. / Li, L., Mu, X., Li, S., & Peng, H. A // IEEE Access, 2020, 8, 139110 —139120. <https://doi.org/10.1109/access.2020.3011028>.
3. Rajendran A. Real Time Attendance Entry Using Face Detection and Recognition. / A. Rajendran, T. Happila, V. Saminathan, P.K. Ghibitha Bebin, K. J. Jenisha, K. Sudha. // International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), 2023. — Pp.806—810.
4. Khan N. A review on moving object detection and tracking in video surveillance. / Khan, N.,& Rehman, A. //Multimedia Tools and Applications, 2018, 77(22), 29835-29857. doi:10.1007/s11042-018-6979-9.
5. Розпізнавання облич як метод для контролю доступу та обліку робочого часу. [Електронний ресурс]. — Режим доступу: <https://supremainc.com.ua/galuzevi-rishennja-suprema/rozpiznavannja-oblych-dlja-kontrolju-dostupu-ta-obliku-robochogo-chasu/>
6. Кушнір Н.О. Використання згорткових нейронних мереж у задачах розпізнавання та класифікації об'єктів зображень / Н.О. Кушнір, Т.М. Локтікова, А.В. Морозов, В.О. Юрченко // Технічна інженерія №1 (89), 2022. — С. 93 —100.
7. Du Y., Wang W., Wang L. Hierarchical recurrent neural network for skeleton based action recognition. // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015. — Vol. 7. №. 12. — P. 1110–1118.
8. Дигодій В. І. Нейромережева технологія виявлення і розпізнавання людей у відеопотоці. / В. І. Дигодій, Т. Б. Мартинюк, М. А. Очуров // Матеріали ЛШ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ-2024)». — Електрон. текст. дані. —

Вінниця, 2024. — Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2024/paper/view/19561/16206>.

9. Заяць В. М. Методи розпізнавання образів. Навч. посібник. / В. М. Заяць, Р. М. Камінський. — Львів, видав. Національного університету «Львівська політехніка», 2004. — 176 с.

10. Face Image Analysis With Convolutional Neural Networks [Електронний ресурс] — режим доступу: <https://www.freidok.unifreiburg.de/fedora/objects/freidok:4835/datastreams/FILE1/content>.

11. Browne M. Convolutional Neural Networks for Image Processing: An Application in Robot Vision, Lecture Notes in Computer Science. / M. Browne, S. Ghidary // Publisher: Springer Berlin Heidelberg, 2003, Volume 293. — Pp. 641 — 652.

12. 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.

13. Синєглазов В. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення [Електронний ресурс] / В. Синєглазов, О. Чумаченко // «Інформаційні технології та комп'ютерне моделювання» : матеріали ст. Міжнар. наук.-практ. конф., м. ІваноФранківськ, 15 — 20 трав. 2017 р. — Івано-Франківськ, 2017. — С. 274 — 277.

14. Мартинюк Т. Б. Системні структури для багатооперандної обробки векторних даних [Текст] : монографія / Т. Б. Мартинюк, В. П. Кожем'яко. — Вінниця : УНІВЕРСУМ-Вінниця, 2007. — 116 с.

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

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

17. 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.

18. 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.

19. 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>.

20. Redmon, J. You only look once: Unified, real-time object detection. / J. Redmon, S. Divvala, R. Girshick & A. Farhadi // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. — Pp. 779-788.

21. Diwan T. Object detection using YOLO: Challenges, architectural successors, datasets and applications. / T. Diwan, G. Anirudh, JV Tembhurne. // Multimedia Tools and Applications, 2023. — Springer — 26 p.

22. Квантові перетворювачі на оптоелектронних логіко-часових середовищах для око-процесорної обробки зображень : монографія / В. П. Кожем'яко, Т. Б. Мартинюк, О. І. Суприган, Д. І. Клімкіна. — Вінниця : УНІВЕРСУМ-Вінниця, 2007. — 126 с.

23. Rosten E., Drummond T. Machine learning for high-speed corner detection // Proc. European Conference on Computer Vision. — 2006. V. 1. — P. 430—443.

24. Кушнір Д. О. Методи пошуку та розпізнавання об'єктів у відеозображеннях на мобільній платформі IOS в реальному часі / Д. О. Кушнір, Я. С. Парамуд // Комп'ютерні системи та мережі. — Львів : Видавництво Львівської політехніки, 2019. — Том 1. № 1. — С. 24–34.

25. Система ідентифікації осіб LFIS Gemalto (Live Face Identification System, біометрична система розпізнавання обличчя) — Thales Group [Електронний ресурс]. — Режим доступу: <https://novations.ua/lfis-gemalto-thales/>.

26. Andrew G. Howard, Menglong Zhu, Bo Chen and others. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision // arXiv. — 2017, 17 April.
27. Kumar, A., et al. LUVLi Face Alignment: Estimating Landmarks' Location, Uncertainty, and Visibility Likelihood. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020. — Pp. 8233 — 8243.
28. Азаров О. Д. Прикладне програмування у комп'ютерних мережах [Текст] : навчальний посібник / О. Д. Азаров, О. І. Черняк., Л. А. Савицька — Вінниця : ВНТУ, 2016. — 130 с.
29. Turati C. et al. Newborns' face recognition: Role of inner and outer facial features // Child development. — 2006. — Т. 77. №. 2. — С. 297—311.
30. Кветний Р. Н. Комп'ютерне моделювання систем та процесорів / І. В. Богач, О. М. Шушура, О. Ю. Софіна — Вінниця: ВНТУ, 2012. — 193 с.
31. Веретельник Д. М. Детекція і розпізнавання осіб з використанням відкритого комп'ютерного зору OpenCV / Д.М. Веретельник, Е.А. Норкене, Д. М. Бочаров // Інформатика, управляючі системи, математичне та комп'ютерне моделювання, 2020. — С. 325 — 330.
32. Kiranyaz S. Operational neural networks. / S. Kiranyaz, T.Ince, , A. Iosifidis, M.Gabbouj, // Neural Computing and Applications. — 2020. [Електронний ресурс]. — Режим доступу: <https://doi.org/10.1007/s00521-020-04780-3>.
33. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.
34. 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..

" "

2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
Нейромережева технологія виявлення і розпізнавання людей у відеопотоці
08-54.МКР.005.00.000.ТЗ

Науковий керівник: д.т.н., професор

_____ Мартинюк Т. Б.

Студент групи 1КІ-22м

_____ Дигодій В. І.

Вінниця 2023

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 нейромережових технологій виявлення і розпізнавання людей у відеопотоці.

4.2 Вивчити принцип роботи та особливості систем розпізнавання людей.

4.3 Запропонувати метод розпізнавання людей в відеопотоці.

4.4 Розробити нейромережеву технологію виявлення і розпізнавання людей у відеопотоці.

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 людей у відеопотоці	26.09.2023р.	30.09.2023р.	Розділ 1
2	Розробка нейромережевої технології виявлення і розпізнавання людей у відеопотоці	01.10.2023р.	04.10.2023р.	Розділ 2
3	Розробка програми виявлення і розпізнавання людей у відеопотоці	05.10.2023р.	18.10.2023р.	Розділ 3
4	Перевірка працездатності програмного продукту	19.10.2023р.	01.11.2023р.	Розділ 4
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання людей у відеопотоці	02.11.2023р.	15.11.2023р.	Розділ 5
6	Апробація та впровадження результатів дослідження	16.11.2023р.	18.11.2023р.	Тези доповіді
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	19.11.2023р.	25.11.2023р.	ПЗ, графічний матеріал
8	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	26.11.2023р.	08.12.2023р.	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

Блок-схема алгоритму інформації про обличчя на форму

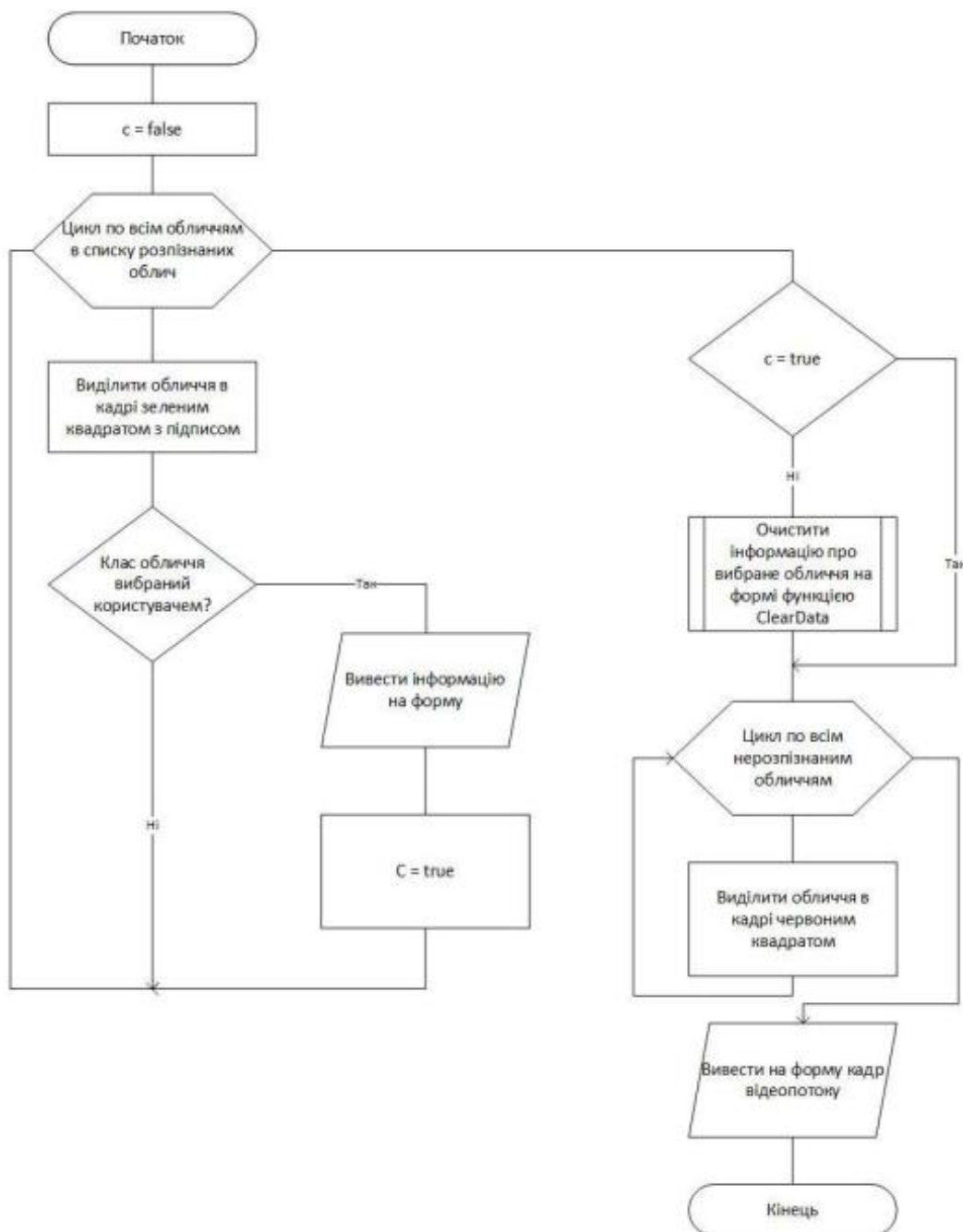


Рисунок Б.1 — Блок-схема алгоритму інформації про обличчя на форму

ДОДАТОК В

Блок-схема алгоритму формування списків розпізнаних та нерозпізнаних обличь

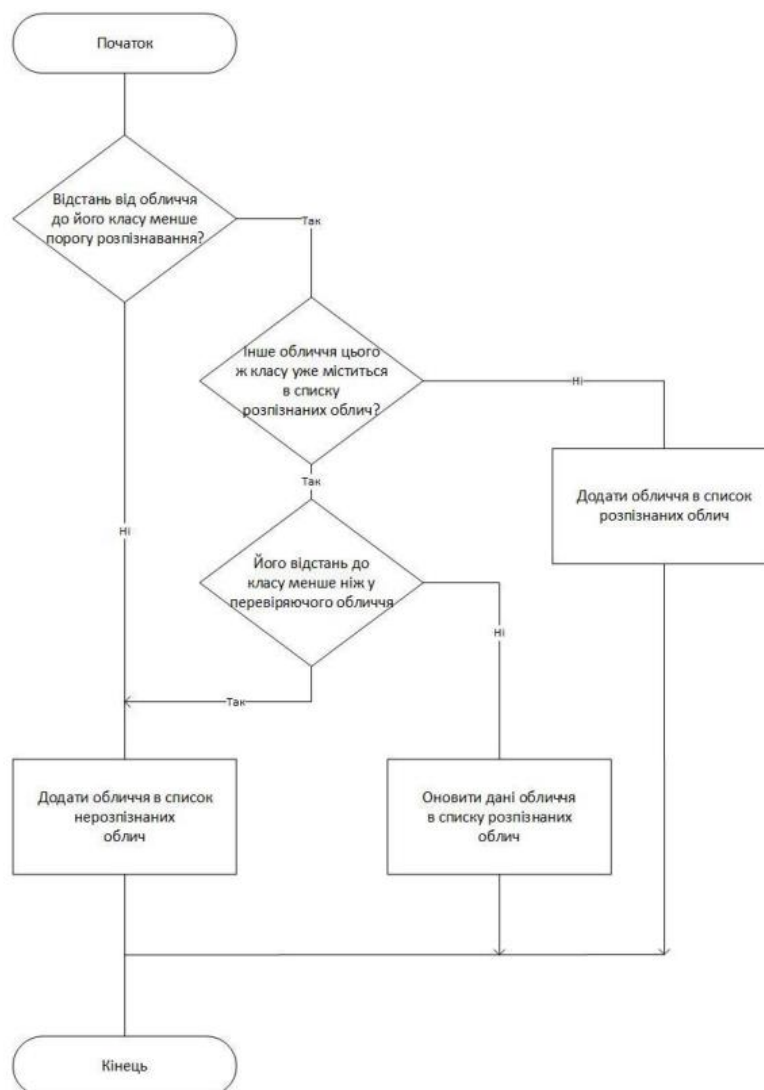


Рисунок В.1 — Блок-схема алгоритму формування списків розпізнаних та нерозпізнаних обличь

ДОДАТОК Г

UML-діаграма класів програмного засобу

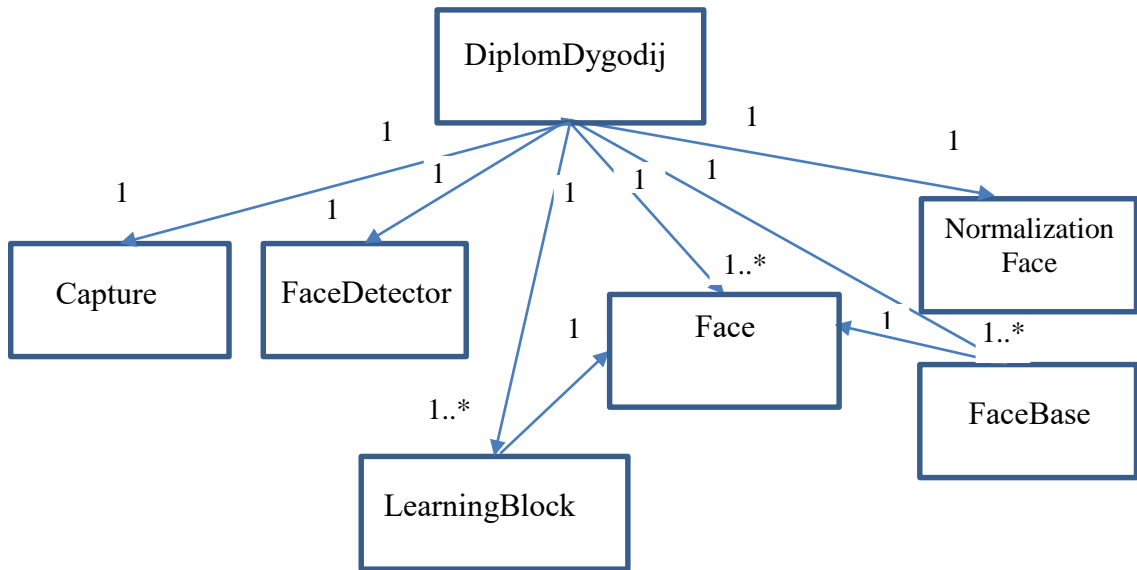


Рисунок Г.1 — UML-діаграма класів програмного засобу

ДОДАТОК Д

Лістинг програми з тренування нейромережі

```

import argparse
#pytorch
from concurrent.futures import thread
from xmlrpc.client import Boolean
from sqlalchemy import null
import torch
from torchvision import transforms
from threading import Thread
#other lib
import sys
import numpy as np
import os
import cv2
import shutil
sys.path.insert(0, "yolov5_face")
from models.experimental import attempt_load
from utils.datasets import letterbox
from utils.general import check_img_size, non_max_suppression_face, scale_coords
# Check device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Get model detect
## Case 1:
# model = attempt_load("yolov5_face/yolov5s-face.pt", map_location=device)
## Case 2:
model = attempt_load("yolov5_face/yolov5n-0.5.pt", map_location=device)
# Get model recognition
## Case 1:
from insightface.insight_face import iresnet100
weight = torch.load("insightface/resnet100_backbone.pth", map_location = device)
model_emb = iresnet100()
## Case 2:
# from insightface.insight_face import iresnet18
# weight = torch.load("insightface/resnet18_backbone.pth", map_location = device)
# model_emb = iresnet18()
model_emb.load_state_dict(weight)
model_emb.to(device)
model_emb.eval()
face_preprocess = transforms.Compose([
    transforms.ToTensor(), # input PIL => (3,56,56), /255.0
    transforms.Resize((112, 112)),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
])

```

```

    ])
def resize_image(img0, img_size):
    h0, w0 = img0.shape[:2] # orig hw
    r = img_size / max(h0, w0) # resize image to img_size
    if r != 1: # always resize down, only resize up if training with augmentation
        interp = cv2.INTER_AREA if r < 1 else cv2.INTER_LINEAR
        img0 = cv2.resize(img0, (int(w0 * r), int(h0 * r)), interpolation=interp)
    imgsz = check_img_size(img_size, s=model.stride.max()) # check img_size
    img = letterbox(img0, new_shape=imgsz)[0]
    # Convert
    img = img[:, :, ::-1].transpose(2, 0, 1).copy() # BGR to RGB, to 3x416x416
    img = torch.from_numpy(img).to(device)
    img = img.float() # uint8 to fp16/32
    img /= 255.0 # 0 - 255 to 0.0 - 1.0
    return img
def get_face(input_image):
    # Parameters
    size_convert = 256
    conf_thres = 0.4
    iou_thres = 0.5
    # Resize image
    img = resize_image(input_image.copy(), size_convert)
    # Via yolov5-face
    with torch.no_grad():
        pred = model(img[None, :])[0]
    # Apply NMS
    det = non_max_suppression_face(pred, conf_thres, iou_thres)[0]
    bboxes = np.int32(scale_coors(img.shape[1:], det[:, :4], input_image.shape).
round().cpu().numpy())
    return bboxes
def get_feature(face_image, training = True):
    # Convert to RGB
    face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2RGB)
    # Preprocessing image BGR
    face_image = face_preprocess(face_image).to(device)
    # Via model to get feature
    with torch.no_grad():
        if training:
            emb_img_face = model_emb(face_image[None, :])[0].cpu().numpy()
        else:
            emb_img_face = model_emb(face_image[None, :]).cpu().numpy()
    # Convert to array
    images_emb = emb_img_face/np.linalg.norm(emb_img_face)
    return images_emb
def read_features(root_feature_path):

```

```

try:
    data = np.load(root_feature_path + ".npz", allow_pickle=True)
    images_name = data["arr1"]
    images_emb = data["arr2"]
    return images_name, images_emb
except:
    return null
def training(full_training_dir, additional_training_dir,
            faces_save_dir, features_save_dir, is_add_user):
    # Init results output
    images_name = []
    images_emb = []
    # Check mode full training or additional
    if is_add_user == True:
        source = additional_training_dir
    else:
        source = full_training_dir
    # Read train folder, get and save face
    for name_person in os.listdir(source):
        person_image_path = os.path.join(source, name_person)
        # Create path save person face
        person_face_path = os.path.join(faces_save_dir, name_person)
        os.makedirs(person_face_path, exist_ok=True)
        for image_name in os.listdir(person_image_path):
            if image_name.endswith(("png", 'jpg', 'jpeg')):
                image_path = person_image_path + f"/{image_name}"
                input_image = cv2.imread(image_path) # BGR
                # Get faces
                bboxes = get_face(input_image)
                # Get boxes
                for i in range(len(bboxes)):
                    # Get number files in person path
                    number_files = len(os.listdir(person_face_path))
                    # Get location face
                    x1, y1, x2, y2 = bboxes[i]
                    # Get face from location
                    face_image = input_image[y1:y2, x1:x2]
                    # Path save face
                    path_save_face = person_face_path + f"/{number_files}.jpg"
                    # Save to face database
                    cv2.imwrite(path_save_face, face_image)
                    # Get feature from face
                    images_emb.append(get_feature(face_image, training=True))
                    images_name.append(name_person)
    # Convert to array

```

```

images_emb = np.array(images_emb)
images_name = np.array(images_name)
features = read_features(features_save_dir)
if features == null or is_add_user== False:
    pass
else:
    # Read features
    old_images_name, old_images_emb = features
    # Add feature and name of image to feature database
    images_name = np.hstack((old_images_name, images_name))
    images_emb = np.vstack((old_images_emb, images_emb))
    print("Update feature!")
# Save features
np.savez_compressed(features_save_dir,
                    arr1 = images_name, arr2 = images_emb)
# Move additional data to full train data
if is_add_user == True:
    for sub_dir in os.listdir(additional_training_dir):
        dir_to_move = os.path.join(additional_training_dir, sub_dir)
        shutil.move(dir_to_move, full_training_dir, copy_function = shutil.copypath)
def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--full-training-dir', type=str, default='./database/full-training-
datasets/', help='dir folder full training')
    parser.add_argument('--additional-training-dir', type=str,
default='./database/additional-training-datasets/', help='dir folder additional training')
    parser.add_argument('--faces-save-dir', type=str, default='./database/face-datasets/',
help='dir folder save face features')
    parser.add_argument('--features-save-dir', type=str,
default='./static/feature/face_features', help='dir folder save face features')
    parser.add_argument('--is-add-user', type=bool, default=True, help='Mode add user
or full training')
    opt = parser.parse_args()
    return opt
def main(opt):
    training(**vars(opt))
if __name__ == "__main__":
    opt = parse_opt()
    main(opt)

```


ДОДАТОК Е

Фрагмент програмного коду

```

#pytorch
import torch
from torchvision import transforms
#other lib
import sys
import numpy as np
import os
import cv2
from insightface.insight_face import iresnet100
from PIL import Image
sys.path.insert(0, "yolov5_face")
from models.experimental import attempt_load
from utils.datasets import letterbox
from utils.general import check_img_size, non_max_suppression_face, scale_coords
# Check device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Get model
# model = attempt_load("yolov5_face/yolov5m-face.pt", map_location=device)
model = attempt_load("yolov5_face/yolov5n-0.5.pt", map_location=device)
weight = torch.load("insightface/16_backbone.pth", map_location = device)
model_emb = iresnet100()
model_emb.load_state_dict(weight)
model_emb.to(device)
model_emb.eval()
face_preprocess = transforms.Compose([
    transforms.ToTensor(), # input PIL => (3,56,56), /255.0
    transforms.Resize((112, 112)),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
])
def resize_image(img0, img_size):
    h0, w0 = img0.shape[:2] # orig hw
    r = img_size / max(h0, w0) # resize image to img_size
    if r != 1: # always resize down, only resize up if training with augmentation
        interp = cv2.INTER_AREA if r < 1 else cv2.INTER_LINEAR
        img0 = cv2.resize(img0, (int(w0 * r), int(h0 * r)), interpolation=interp)
    imgsiz = check_img_size(img_size, s=model.stride.max()) # check img_size
    img = letterbox(img0, new_shape=imgsiz)[0]
    # Convert
    img = img[:, :, ::-1].transpose(2, 0, 1).copy() # BGR to RGB, to 3x416x416
    img = torch.from_numpy(img).to(device)
    img = img.float() # uint8 to fp16/32

```

```

    img /= 255.0 # 0 - 255 to 0.0 - 1.0
    return img
def get_face(input_image):
    # Parameters
    size_convert = 640
    conf_thres = 0.4
    iou_thres = 0.5
    # Resize image
    img = resize_image(input_image.copy(), size_convert)
    # Via yolov5-face
    with torch.no_grad():
        pred = model(img[None, :])[0]
    # Apply NMS
    det = non_max_suppression_face(pred, conf_thres, iou_thres)[0]
    bboxes = np.int32(scale_coords(img.shape[1:], det[:, :4],
input_image.shape).round().cpu().numpy())
    return bboxes
def get_feature(face_image, training = True):
    # Convert to RGB
    face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2RGB)
    # Preprocessing image BGR
    face_image = face_preprocess(face_image).to(device)
    # Via model to get feature
    with torch.no_grad():
        if training:
            emb_img_face = model_emb(face_image[None, :])[0].cpu().numpy()
        else:
            emb_img_face = model_emb(face_image[None, :]).cpu().numpy()
    # Convert to array
    images_emb = emb_img_face/np.linalg.norm(emb_img_face)
    return images_emb
def read_features(root_feature_path = "static/feature/face_features.npz"):
    data = np.load(root_feature_path, allow_pickle=True)
    images_name = data["arr1"]
    images_emb = data["arr2"]
    return images_name, images_emb
def training(input_image, name_persion):
    # input_image = cv2.imread(path_image)
    # Path to save face
    path_database = "faces_database/"
    path_persion = os.path.join(path_database, name_persion)
    # Create dir
    os.makedirs(path_persion, exist_ok=True)
    images_name = []
    images_emb = []

```

```

# Get faces
bboxes = get_face(input_image)
# Get boxes
for i in range(len(bboxes)):
    # Get number files in persion path
    number_files = len(os.listdir(path_persion))
    # Get location face
    x1, y1, x2, y2 = bboxes[i]
    # Get face from location
    face_image = input_image[y1:y2, x1:x2]
    # Path save face
    path_save_face = path_persion + f'/{number_files}.jpg'
    # Save to face database
    cv2.imwrite(path_save_face, face_image)
    # Get feature from face
    images_emb.append(get_feature(face_image, training=True))
    images_name.append(name_persion)
# Convert to array
images_emb = np.array(images_emb)
images_name = np.array(images_name)
# Read features
old_images_name, old_images_emb = read_features()
# Add feature and name of image to feature database
new_images_name = np.hstack((old_images_name, images_name))
new_images_emb = np.vstack((old_images_emb, images_emb))
# Save features
path_features = "static/feature/"
np.savez_compressed(path_features + "face_features",
                    arr1 = new_images_name, arr2 = new_images_emb)
def recognition(image):
    # Get faces
    bboxes = get_face(image)
    # Get boxes
    for i in range(len(bboxes)):
        # Get location face
        x1, y1, x2, y2 = bboxes[i]
        # Get face from location
        face_image = image[y1:y2, x1:x2]
        # Get feature from face
        query_emb = (get_feature(face_image, training=False))
        print(query_emb.shape)
        # Read features
        images_names, images_embs = read_features()
        scores = (query_emb @ images_embs.T)[0]
        id_min = np.argmax(scores)

```

```

score = scores[id_min]
name = images_names[id_min]
if score < 0.2:
    caption= "UN_KNOWN"
else:
    caption = f"{name.split('_')[0].upper()}:{score:.2f}"
t_size = cv2.getTextSize(caption, cv2.FONT_HERSHEY_PLAIN, 2, 2)[0]
cv2.rectangle(image, (x1, y1), (x2, y2), (0, 146, 230), 3)
cv2.rectangle(
    image, (x1, y1), (x1 + t_size[0], y1 + t_size[1]), (0, 146, 230), -1)
cv2.putText(image, caption, (x1, y1 + t_size[1]),
cv2.FONT_HERSHEY_PLAIN, 2, [255, 255, 255], 2)
return image
if __name__=="__main__":
    ""Traning""
    # Input
    path_image = "train_image.jpg"
    name_persion = "Châu"
    image = cv2.imread(path_image)
    # Training
    training(image, name_persion)
    ""Recognition""
    # Input
    path_query = "test_image.jpg"
    image = cv2.imread(path_query)
    # Regnition
    result = recognition(image)
    cv2.imwrite("result.jpg", result)
    # Output
    cv2.imshow("Result", result)
    cv2.waitKey(0) # waits until a key is pressed
    cv2.destroyAllWindows() # destroys the window showing image

```

