

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Методи та програмно-апаратні засоби автоматизованої ідентифікації
і сортування медіаконтенту

Виконав: студент(ка) 6 курсу, групи СІм-61
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

	<u>Музичук А.А.</u> (підпис) (прізвище та ініціали)
Керівник	<u>Шингера Н.Я.</u> (підпис) (прізвище та ініціали)
Нормоконтроль	<u>Луцик Н.С.</u> (підпис) (прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (підпис) (прізвище та ініціали)
Рецензент	<u>Деркач М.В.</u> (підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«17» листопада 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Музичуку Андрію Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та програмно-апаратні засоби автоматизованої ідентифікації і сортування медіаконтенту

Керівник роботи Шингера Наталія Ярославівна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 14 » листопада 2025 року № 4/7-987

2. Термін подання студентом завершеної роботи 16 грудня 2025 р.

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1 Аналіз сучасних технологій і практик

2 Розробка програмного забезпечення для автоматизованого сортування медіаконтенту

3 Практична реалізація системи

4 Охорона праці та безпека в надзвичайних ситуаціях

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження.

2. Завдання, об'єкт і предмет, наукова новизна і практична цінність дослідження.

3.

4.

5.

6.

7.

8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці</i>	<i>Осухівська Г.М., зав.каф. КС</i>	<i>04.12.2025</i>	
<i>Безпека в надзвичайних ситуаціях</i>	<i>Стручок В.С., ст. викладач каф. ОХ</i>		

7. Дата видачі завдання 17.11.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Огляд літературних джерел. Постановка завдання на кваліфікаційну роботу.</i>	<i>17.11.2025р. – 23.11.2025р.</i>	<i>Викон.</i>
2.	<i>Розробка програмного забезпечення для автоматизованого сортування медіаконтенту</i>	<i>24.11.2025р. – 02.12.2025р.</i>	<i>Викон.</i>
3.	<i>Практична реалізація системи</i>	<i>03.12.2025р. – 10.12.2025р.</i>	<i>Викон.</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>04.12.2025р. – 10.12.2025р.</i>	<i>Викон.</i>
5.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>11.12.2025р. – 19.12.2025р.</i>	<i>Викон.</i>
6.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>16.12.2025р.</i>	<i>Викон.</i>
7.	<i>Захист кваліфікаційної роботи магістра</i>	<i>26.12.2025р.</i>	<i>Викон.</i>

Студент

(підпис)*Музичук А.А.*

(прізвище та ініціали)

Керівник роботи

(підпис)*Шингера Н.Я.*

(прізвище та ініціали)

АНОТАЦІЯ

Музичук А.А. Методи та програмно-апаратні засоби автоматизованої ідентифікації і сортування медіаконтенту: робота на здобуття кваліфікаційного ступеня магістра: спец. 123 — комп'ютерна інженерія / наук.кер. Шингера Н.Я. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: комп'ютерна система, медіаконтент, комп'ютерний зір, нейронна мережа, YOLO, Python, H.265, класифікація відео.

Кваліфікаційна робота присвячена розробці методів та програмних засобів для автоматизованої ідентифікації та сортування відеоконтенту. У роботі проведено аналіз сучасних алгоритмів комп'ютерного зору та стандартів компресії відеоданих для довготривалого зберігання. Розроблено програму мовою Python із використанням бібліотек OpenCV та Ultralytics.

Основою системи ідентифікації обрано нейронну мережу архітектури YOLO11, яка забезпечує детекцію об'єктів у реальному часі. Реалізовано алгоритм логічного мапінгу для категоризації відео за змістом та підсистему автоматичного транскодування файлів у формати H.264/H.265 за допомогою FFmpeg. Результати роботи дозволяють автоматизувати процес впорядкування великих медіаархівів та суттєво зменшити обсяг займаного дискового простору без втрати візуальної якості.

ANNOTATION

Muzychuk A.A. Methods and hardware–software tools for automated identification and sorting of media content. Master’s Graduation Thesis: speciality 123 — Computer engineering / supervisor Shynhera N.Y. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: computer system, media content, computer vision, neural network, YOLO, Python, H.265, video classification.

The qualification paper is devoted to the development of methods and software tools for the automated identification and sorting of video content. The paper analyzes modern computer vision algorithms and video data compression standards for long-term storage. A software system was developed using the Python programming language and the OpenCV and Ultralytics libraries.

The YOLO11 neural network architecture was selected as the core of the identification system, ensuring real-time object detection. A logical mapping algorithm for content-based video categorization and a subsystem for automatic file transcoding into H.264/H.265 formats using FFmpeg have been implemented. The results of the study allow for automating the organization of large media archives and significantly reducing disk space usage without compromising visual quality.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ І ПРАКТИК.....	12
1.1. Область застосувань технології.....	12
1.2. Сучасні системи комп'ютерного зору	15
1.3. Актуальні системи комп'ютерного зору	17
1.4. Принципи функціонування згорткових нейронних мереж у задачах автоматизованого аналізу зображень	18
1.5. Аналіз сучасних методів та стандартів компресії відеоданих для систем довготривалого зберігання	20
1.6. Особливості компресії відеопотоку для задач комп'ютерного зору та машинної ідентифікації об'єктів	22
1.7. Висновки до розділу 1	23
РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО СОРТУВАННЯ МЕДІАКОНТЕНТУ	25
2.1. Загальна архітектура системи.....	25
2.2. Приклад використання та функціональні вимоги	27
2.3. Обґрунтування вибору моделі комп'ютерного зору для підсистеми ідентифікації відеоконтенту	28
2.4. Вибір мови програмування та середовища розробки.....	35
2.5. Опис алгоритму роботи програми.....	36
2.6. Розробка алгоритму прийняття рішень та логіки класифікації	38
2.7. Висновки до розділу 2	39
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ	41
3.1. Розгортання середовища розробки та його налаштування.....	41

3.2.	Розгортання моделі YOLO11 та перевірка її працездатності.....	44
3.3.	Програмна реалізація функціоналу ідентифікації відеоконтенту.....	46
3.4.	Реалізація підсистеми транскодування та оптимізації зберігання відеоданих.	51
3.5.	Тестування та порівняльний аналіз ефективності роботи системи	53
3.6.	Транскодування відео за допомогою розробленої системи	55
3.7.	Висновки до Розділу 3	56
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ		58
4.1.	Охорона праці.....	58
4.2.	Безпека в надзвичайних ситуаціях	61
ВИСНОВКИ.....		65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		66
Додаток А Тези конференцій		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CV (англ. Computer Vision) Комп'ютерний зір

CNN (англ. convolutional neural network) Згорткові нейронні мережі

VCM (англ. Video Coding for Machines) Відеокompresія для комп'ютерних систем

CRF (англ. Constant Rate Factor) Постійне значення оцінки

ВСТУП

Актуальність роботи. У сучасному цифровому просторі спостерігається експоненційне зростання обсягів мультимедійних даних, що генеруються смартфонами, системами відеонагляду та автомобільними реєстраторами. Накопичення масивів неструктурованого відеоконтенту створює проблему "інформаційного шуму", коли пошук необхідної інформації унеможлиблюється через відсутність семантичної розмітки та метаданих. Традиційні методи ручного сортування є неефективними через великі часові затрати, а хмарні рішення часто обмежені вартістю або політикою конфіденційності. Тому створення автономних систем, здатних "розуміти" зміст відео та автоматично його впорядковувати, набуває критичного значення.

Питанням комп'ютерного зору та детекції об'єктів присвячено праці багатьох закордонних та вітчизняних науковців, зокрема J. Redmon, A. Bochkovskiy, G. Jocher, R. Szeliski, Y. LeCun, C.-Y. Wang та ін. У їхніх роботах закладено фундамент архітектур згорткових нейронних мереж (CNN) та алгоритмів серії YOLO. Проте більшість існуючих досліджень фокусуються виключно на точності детекції, залишаючи поза увагою комплексні питання оптимізації зберігання відсортованих даних. У цій роботі пропонується підхід, що поєднує швидкісну ідентифікацію об'єктів на базі новітньої архітектури YOLO11 із технологіями ефективного кодування відео (H.265/HEVC). Таке поєднання дозволяє не лише структурувати архів, а й суттєво зменшити вимоги до апаратних ресурсів зберігання. Тому розробка та дослідження методів автоматизованого сортування відеоконтенту є актуальною науково-прикладною задачею.

Метою кваліфікаційної роботи є підвищення ефективності обробки та зберігання великих масивів неструктурованих відеоданих шляхом розробки комп'ютеризованої системи для автоматизованої ідентифікації, семантичної класифікації та адаптивного транскодування медіаконтенту.

Завдання кваліфікаційної роботи:

- Провести аналітичний огляд сучасних методів компресії відеоданих (H.264,

H.265/HEVC) та алгоритмів комп'ютерного зору для детекції об'єктів;

- Обґрунтувати вибір програмних засобів та архітектури нейронної мережі для забезпечення роботи системи в режимі реального часу;
- Розробити алгоритм автоматизованого сортування відеофайлів, що поєднує методи нейромережевої детекції з логічним мапінгом категорій;
- Програмно реалізувати підсистему транскодування для оптимізації обсягу архіву без втрати інформативності;
- Провести експериментальне дослідження розробленої системи, порівняти ефективність моделей YOLOv8 та YOLO11, а також оцінити ступінь компресії даних.

Відповідно до мети та завдань кваліфікаційної роботи визначено її об'єкт та предмет.

Об'єкт дослідження: процес автоматизованої обробки та аналізу відеопотоків у комп'ютерних системах.

Предмет дослідження: методи та програмні засоби ідентифікації об'єктів на відеозображеннях з використанням згорткових нейронних мереж та алгоритми оптимізації зберігання мультимедійних даних.

Методи дослідження: Для розв'язання поставлених завдань у роботі використано відповідні методи:

- методи системного аналізу для побудови загальної архітектури програмного комплексу та моделювання взаємодії між модулями детекції, логічної обробки та архівації даних;
- методи машинного навчання та комп'ютерного зору при застосуванні та налаштуванні згорткових нейронних мереж архітектури YOLO для розпізнавання образів у динамічному відеопотоці;
- теорія алгоритмів для розробки евристичних алгоритмів прийняття рішень при класифікації контенту та оптимізації частоти обробки кадрів;
- експериментальні методи та порівняльний аналіз для оцінки ефективності стиснення відеоданих кодеками H.264/H.265 та дослідження швидкодії різних версій нейронних мереж.

Наукова новизна дослідження. Вперше запропоновано метод комплексної оптимізації медіаархівів, який, на відміну від існуючих рішень, інтегрує високошвидкісну детекцію об'єктів на базі нейронної мережі YOLO11 з адаптивним транскодуванням даних у формат H.265/HEVC. Це дозволяє здійснювати автоматичну структурування відеоконтенту з одночасним зменшенням обсягу займаного дискового простору без втрати інформативності.

Удосконалено алгоритм семантичної класифікації відеопотоку шляхом розробки логічного модуля агрегації метаданих, що забезпечує перетворення множини розрізнених класів об'єктів, виявлених нейромережею, у фіксовану структуру категорій архіву, підвищуючи точність прийняття рішень системою.

Практичне значення результатів кваліфікаційної роботи. Створено та програмно реалізовано автоматизовану систему сортування медіаконтенту, що дозволяє виключити необхідність ручної обробки великих масивів відеоданих та значно скоротити час на їх впорядкування. Запропонований програмний засіб може бути ефективно використаний як для персональних потреб при впорядкуванні домашніх архівів, так і в професійній діяльності для сортування записів з відеореєстраторів або систем безпеки.

Публікації. Результати дослідження апробовано на XIII науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» та XIV міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій», у вигляді тез конференцій.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, 4 розділів, висновків, списку використаних джерел та додатку (-ів).

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ І ПРАКТИК

1.1. Область застосувань технології

Технології стиснення медіа й системи комп'ютерного зору утворюють взаємодоповнювальну систему. Перші оптимізують збереження й передачу великих обсягів візуальної інформації, другі надають сенс цій інформації, автоматизуючи її інтерпретацію, індексацію й прийняття рішень. Розробка алгоритмів стиснення відбувається не лише з метою мінімізації займаного місця на накопичувачах, а й для зменшення вимог до пропускної здатності мереж, скорочення енергоспоживання при передачі і обробці, підвищення доступності контенту на мобільних і edge-пристроях та забезпечення прийняттого співвідношення якості/затримка у режимі реального часу. Водночас компресія впливає на корисність даних для подальшого автоматизованого аналізу. Від ступеня втрат при кодуванні залежить точність детекції, класифікації та інших алгоритмів машинного зору, тому сучасні дослідження й інженерні рішення прагнуть оптимізувати компроміс між розміром бітрейту і збереженням семантичної інформації кадрів.

Системи комп'ютерного зору застосовуються у великому спектрі прикладних задач, як на рисунку 1.1, де необхідне автоматичне «розуміння» візуального ряду.

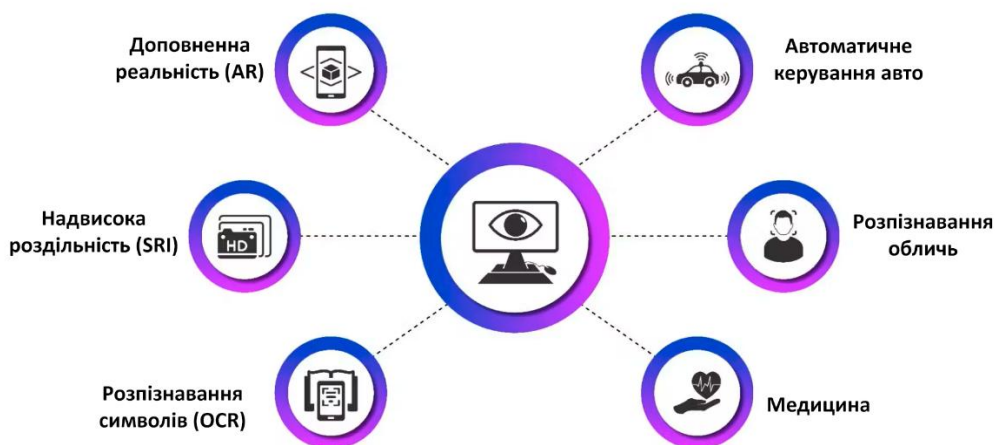


Рис. 1.1. Сфери застосування комп'ютерного зору

У прикладних доменах відеоспостереження та безпеки CV-мережі забезпечують виявлення й відслідковування об'єктів, розпізнавання подій і аномалій, наприклад, переповнення приміщення, залишені предмети, підозріла активність, що дозволяє скоротити потребу в ручному моніторингу та підвищити оперативність реагування. У транспортній індустрії алгоритми детекції та семантичної сегментації лежать в основі систем допомоги водієві й автономного керування. Тут важлива як точність локалізації об'єктів, так і їхня стабільна ідентифікація за умов шуму та стиснення відеопотоку. У медичній діагностиці комп'ютерний зір допомагає автоматизувати аналіз знімків, наприклад, рентген, КТ, ендоскопія, що підвищує швидкість і стандартизацію інтерпретації. У сфері медіаархівів і стримінгу індексація кадрів і автоматичне тегування відео дозволяють реалізувати швидкий пошук за змістом, персоналізовані рекомендації та автоматичну генерацію прев'ю чи субтитрів. Сфера виробничих застосувань може включати в себе контроль якості на конвеєрі, робоча взаємодія з роботами, детектування дефектів, позиціонування та вимірювань у реальному часі. Або використання інструментів CV для витягання смислових ознак із відео та зображень.

Однією з важливих тенденцій сучасної обробки відео є перехід від традиційного «людинозрозумілого» кодування до підходів, орієнтованих на потреби машинного аналізу. Класичні кодеки, такі як H.264 або HEVC, оптимізовані передусім для візуальної якості, тобто для того, щоб відео виглядало чітким для людського ока навіть за сильного стиснення [11,13]. Приклад роботи роботи кодеків зображений на рисунку 1.2

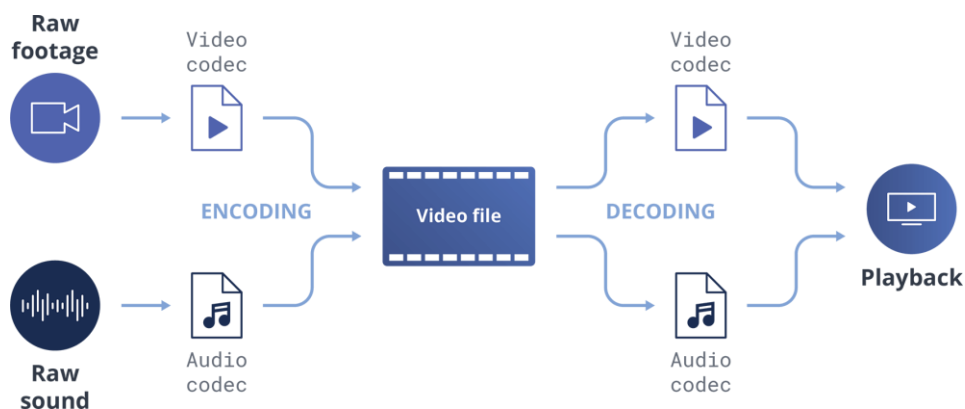


Рис. 1.2 Схема роботи алгоритму кодування

Тому з'являються підходи та стандарти Video Coding for Machines [12,13], які оптимізують кодування з урахуванням подальших задач комп'ютерного зору, а не лише візуальної якості для людини.

Архітектури моделей еволюціонують у бік більш універсальних і масштабованих рішень. Трансформерна парадигма і великі мультимодальні моделі розширюють можливості семантичного розуміння відео і дозволяють працювати із контекстом на довших часових діапазонах.

У зв'язку з обмеженими ресурсами пристроїв розвивається напрямок оптимізації моделей і апаратного прискорення: квантовані моделі, пранераметризація, дистилляція, а також NPUs/TPUs/ASIC для інференсу на edge-рівні, роблять задачі CV доступними в енергоефективних реалізаціях.

Крім технічних і продуктивних аспектів, важливим вектором є інтеграція з інфраструктурою передачі даних і стандартами стримінгу. Поява протоколів і практик для низьколатентної передачі (WebRTC, SRT, LL-HLS) та систем адаптивного бітрейту дозволяє поєднати вимоги реального часу із збереженням якості, необхідної для надійної роботи алгоритмів розпізнавання. Паралельно з ростом точності розпізнавання виникають питання приватності, безпеки й етичності застосування. Технологічний прогрес супроводжується необхідністю розробки механізмів анонімізації, контролю доступу до даних, прозорих політик використання та правових рамок, які гарантують допустиме застосування систем.

У підсумку, область застосувань технологій стиснення й комп'ютерного зору охоплює як інфраструктурні питання такі як, оптимізація зберігання та передачі великих відеоархівів, забезпечення низької затримки в стримінгу, так і прикладні сервіси: моніторинг, діагностика, рекомендації, автоматична індексація. Подальший розвиток цієї галузі залежатиме від одночасного прогресу в моделях інтерпретації зображень, у методах задачо-орієнтованого кодування відео, у стандартах передачі й у доступності апаратного прискорення.

1.2. Сучасні системи комп'ютерного зору

Сучасні системи комп'ютерного зору проектуються як багаторівневі інформаційні конвеєри, що перетворюють необроблений відеопотік або набір зображень у структуровані семантичні описи й дії [14]. На найнижчому рівні цей конвеєр включає отримання та попередню обробку сигналу далі слідує екстракція ознак та побудова глобальних представлень кадрів, потім виявлення й локалізація об'єктів, просторово-часова асоціація і, нарешті, семантична інтерпретація як зображено на рисунку 1.3.



Рис. 1.3. Схема роботи системи CV

Архітектурно такі системи поєднують класичні методи обробки зображень та сучасні модулі машинного навчання, що забезпечують роботу на різних часових і просторових масштабах [15].

Історично початковими будівельними блоками були дескриптори й детектори на основі вручну заданих ознак та класичні класифікатори. Пріоритет у продуктивності і здатності виявляти складні семантичні ознаки перейшов до глибинних нейромереж, відомих як згорткові мережі, які стали стандартом для екстракції просторових ознак із зображень. Для задач детекції об'єктів сформувались дві фундаментальні парадигми: двоетапні підходи, де спочатку генеруються регіон-кандидати, а потім виконується їхнє уточнення, прикладом є Faster R-CNN, та однострімні детектори, які оцінюють положення й класи прямо в одному проході

мережі, наприклад YOLO. Однострінні детектори забезпечують високу швидкість інференсу і тому широко застосовуються у сценаріях реального часу та в обмежених обчислювальних умовах, тоді як двоетапні підходи залишаються стандартом там, де критична максимально можлива точність детекції.

Для відеоаналізу важливо враховувати часову динаміку. Нещодавній підйом архітектур на базі трансформерів у візуальній області дав змогу ефективніше опрацьовувати глобальні залежності в кадрі й між кадрами, що сприяє кращому розумінню сцен із складним контекстом; у багатьох завданнях трансформери або гібридні CNN–Transformer рішення показують конкурентні або кращі результати, ніж класичні підходи.

Інструменти для побудови кінцевих систем охоплюють не тільки моделі детекції/класифікації, але й підсистеми трекінгу й індексації. Для асоціації об'єктів між кадрами застосовуються алгоритми, що поєднують швидкий геометричний трекінг із зовнішньою верифікацією за зовнішніми ознаками. Для задач пошуку схожих кадрів чи витягання репрезентацій широко використовують метричне навчання та сіамські архітектури, які формують embedding-простір візуалізований на рисунку 1.4 [14], у якому знаходження найближчих сусідів здійснюється через індексацію.

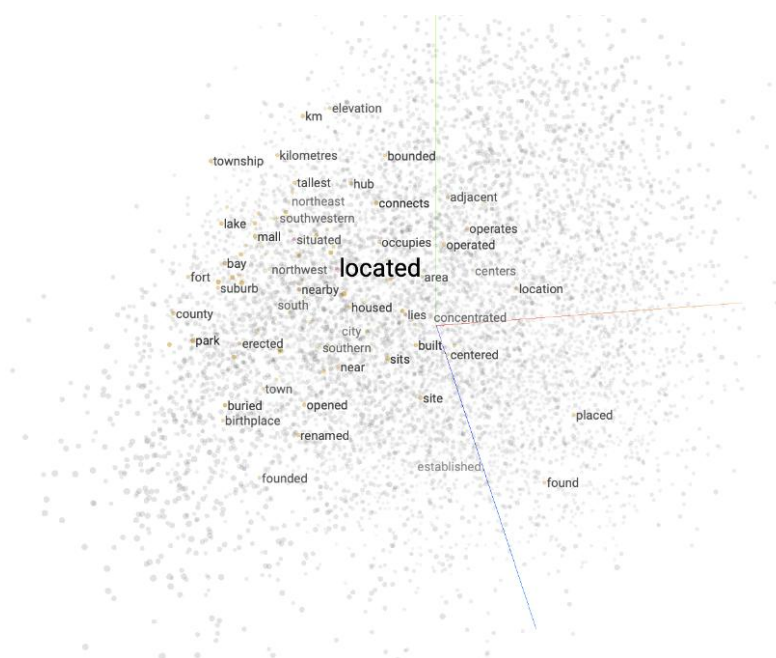


Рис. 1.4. Візуалізація embedding-простору

Одночасно системи семантичної сегментації дозволяють отримувати маски об'єктів/контурів і використовуються там, де потрібна детальна форма об'єктів.

1.3. Актуальні системи комп'ютерного зору

Великі технологічні корпорації та спеціалізовані гравці застосовують CV-системи у широкому спектрі. роздрібні сервіси автоматичної оплати й відслідковування товарних позицій, прикладом є підходи типу Amazon Go, промислова автоматизація та контроль якості на конвеєрах, системи ADAS та автопілотування в автомобілях, а також відеоаналітика у служб безпеки і розумні міські рішення[15]. Реальна індустріальна інтеграція ілюструється прикладами використання комп'ютерного зору великими корпораціями, які поєднують власні хмарні сервіси й апаратні платформи для масштабного розгортання CV-рішень.

Переваги, які дають впровадження сучасних систем комп'ютерного зору для компаній та організацій, можна охарактеризувати через кілька взаємопов'язаних ефектів. Наприклад, автоматизація рутинного моніторингу й аналізу дозволяє знизити експлуатаційні витрати і прискорити реагування на інциденти. Підвищення якості сервісів збільшує залученість користувачів та комерційну вигоду. Аналітика на основі відео відкриває нові бізнес-вимірювання нарешті, застосування CV у безпеці та медицині підвищує рівень точності й стандартизації прийнятих рішень. Важливо зазначити, що продуктивність систем залежить від якості вхідного відео.

У технічній практиці при розгортанні CV-систем ключовими питаннями залишаються оптимізація моделей, вибір місця обчислень, та інтеграція з механізмами зберігання і стиснення відео, щоб зменшити навантаження на канали і архіви. Таким чином, сучасні системи комп'ютерного зору поєднують широкий набір алгоритмів і архітектур, а їх практичне застосування демонструє значний економічний і соціальний ефект[16].

1.4. Принципи функціонування згорткових нейронних мереж у задачах автоматизованого аналізу зображень

Основою сучасної підсистеми комп'ютерного зору для ідентифікації та класифікації медіаконтенту є використання згорткових нейронних мереж (CNN). Цей клас глибоких нейронних мереж продемонстрував найвищу ефективність у задачах розпізнавання образів, оскільки їх архітектура біологічно натхненна будовою зорової кори головного мозку тварин та людини як на рисунку 1.5 [15].

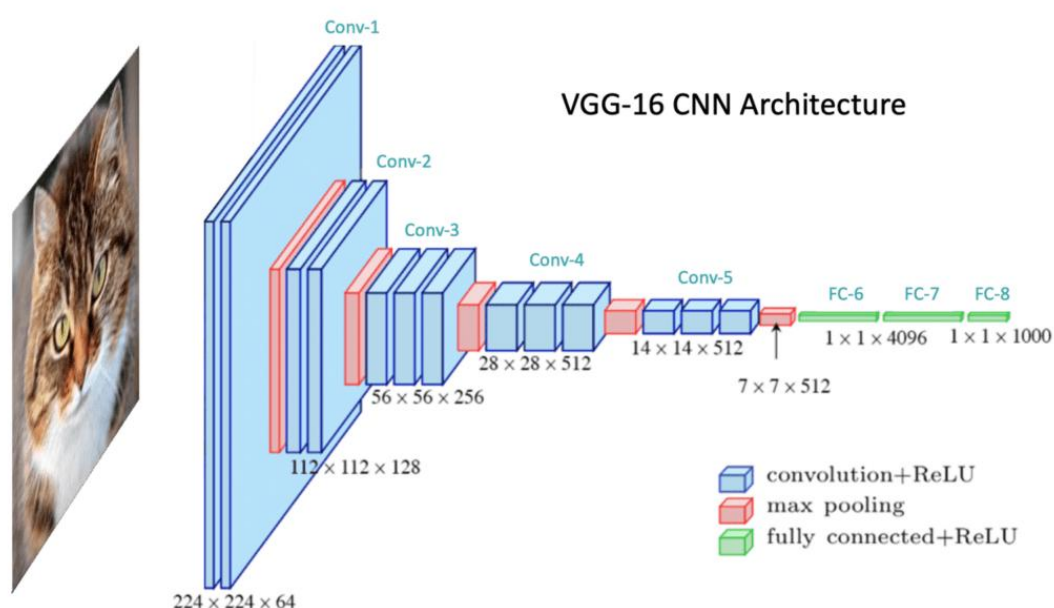


Рис. 1.5. Архітектура згорткової мережі VGG-16

На відміну від класичних алгоритмів машинного навчання, де виділення характерних ознак зображення вимагало ручного конструювання дескрипторів, згорткові мережі здатні автоматично формувати ієрархію абстрактних ознак безпосередньо з необроблених піксельних даних у процесі навчання. Це дозволяє системі адаптуватися до різноманітних типів відеоконтенту та забезпечує високу стійкість до змін освітлення, масштабу та ракурсу зйомки.

Ключовим елементом архітектури CNN, що відрізняє її від повнозв'язних мереж, є операція згортки. Математично цей процес представляє собою проходження невеликого фільтра, який називається ядром згортки, по всій площині вхідного

зображення. Ядро має фіксований розмір, зазвичай 3 на 3 або 5 на 5 пікселів, і містить набір вагових коефіцієнтів, які налаштовуються під час тренування мережі. У кожній позиції ковзного вікна обчислюється скалярний добуток між значеннями пікселів зображення та вагами ядра, що дозволяє сформувати карту ознак. Ця операція дозволяє мережі детектувати локальні патерни, такі як лінії, кути, перепади контрасту або специфічні кольорові переходи, незалежно від їх розташування на кадрі, забезпечуючи властивість трансляційної інваріантності.

Важливою складовою, що слідує за шарами згортки, є шари підвибірки або пулінгу. Їх основним завданням є поступове зменшення просторової розмірності карт ознак, що дозволяє скоротити кількість параметрів мережі та обсяг необхідних обчислень, а також запобігти перенавчанню моделі. Найпоширенішим методом є операція Max Pooling, яка вибирає максимальне значення з локальної ділянки карти ознак. Цей процес дозволяє виділити найбільш значущі, домінуючі ознаки та відфільтрувати шум. Завдяки чергуванню шарів згортки та пулінгу, нейронна мережа здатна переходити від аналізу локальних деталей до розпізнавання глобальних об'єктів. Якщо перші шари реагують на прості геометричні примітиви, то глибші шари формують складні семантичні поняття, такі як "обличчя людини", "автомобіль" або "текст"[15].

Для забезпечення нелінійності, яка необхідна для апроксимації складних залежностей у даних, після кожного згорткового шару застосовується функція активації. Стандартом де-факто у сучасних архітектурах є функція ReLU (Rectified Linear Unit), яка замінює всі від'ємні значення на нуль, залишаючи додатні без змін. Це дозволяє прискорити процес навчання завдяки вирішенню проблеми зникаючого градієнта, характерної для сигмоїдальних функцій, та робить мережу більш чутливою до корисних сигналів. Саме нелінійні перетворення дозволяють CNN розділяти класи об'єктів, які неможливо відокремити лінійно у вихідному просторі ознак.

Завершальним етапом обробки даних у згортковій нейронній мережі є класифікація, яку виконують повнозв'язні шари. На цьому етапі тривимірні карти ознак розгортаються у одновимірний вектор, який подається на вхід класичного перцептрону. Останній шар мережі, як правило, використовує функцію активації

Softmax, яка перетворює вихідні значення нейронів у розподіл ймовірностей приналежності об'єкта до певного класу. Саме цей вихідний вектор ймовірностей є основою для роботи алгоритму сортування медіаконтенту: якщо ймовірність наявності певного об'єкта перевищує встановлений поріг, система приймає рішення про присвоєння відеофайлу відповідної категорії або тегу.

1.5. Аналіз сучасних методів та стандартів компресії відеоданих для систем довготривалого зберігання

Питання ефективного зберігання медіаконтенту є критично важливим етапом у проектуванні систем автоматизованої ідентифікації та сортування. Оскільки результатом роботи системи є не лише метадані про вміст, але й сам відеофайл, який необхідно архівувати.

Протягом останнього десятиліття домінуючим стандартом залишався H.264. Його повсюдне поширення зумовлене високою сумісністю та відносно низькими вимогами до обчислювальних потужностей при кодуванні. Проте в умовах роботи з відео високої роздільної здатності (4K та вище), ефективність H.264 стрімко знижується, що призводить до невиправдано великих обсягів файлів. Це робить його використання у сучасних системах архівування медіаконтенту нераціональним. На зміну йому прийшов стандарт H.265, також відомий як HEVC (High Efficiency Video Coding). Основною перевагою цього алгоритму є здатність забезпечувати ідентичну візуальну якість зображення при зменшенні розміру файлу приблизно на 50% у порівнянні з H.264 згідно таблиці на рисунку 1.6 [17].

Resolution	H.264/AVC Bandwidth required	H.265/HEVC Bandwidth required
480p	1.5 Mbps	0.75 Mbps
720p	3 Mbps	1.5 Mbps
1080p	6 Mbps	3 Mbps
4K	32 Mbps	15 Mbps

Рис. 1.6. Таблиця ресурсоемкості алгоритмів стискання

Такої ефективності вдалося досягти завдяки переходу від фіксованих макроблоків до деревоподібних структур кодування, які можуть динамічно змінювати свій розмір залежно від складності сцени, наприклад як на рисунку 1.7.

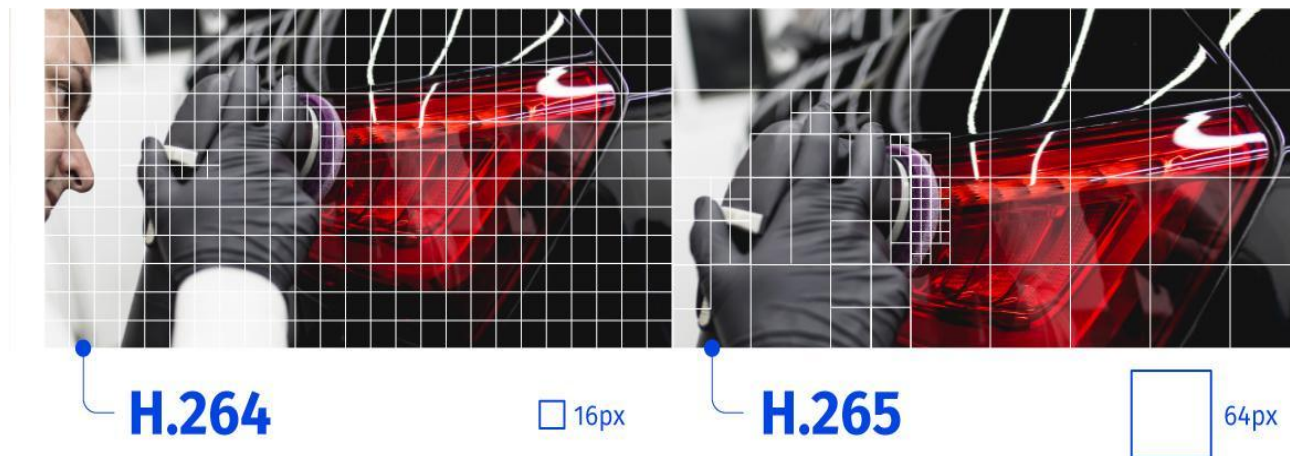


Рис. 1.7. Візуалізація структури блочного кодування

Для розроблюваної системи сортування це означає можливість подвоїти глибину архіву без збільшення фізичного дискового простору. Хоча кодування у H.265 вимагає значно більших обчислювальних ресурсів, сучасні апаратні енкодери, такі як NVENC та Intel QuickSync успішно нівелюють цей недолік, дозволяючи виконувати компресію в реальному часі.

Окремої уваги заслуговує формат AV1 (AOMedia Video 1), розроблений консорціумом технологічних гігантів як відкрита альтернатива пропрієтарним кодекам. AV1 демонструє ще вищу ефективність стиснення, перевершуючи HEVC у середньому на 20-30% у сценаріях високої чіткості. Цей кодек використовує вдосконалені методи передбачення руху та фільтрації, що робить його кандидатом для подальшого зберігання відсортованого матеріалу. Однак, впровадження AV1 у систему може бути обмеженим через високу обчислювальну складність процесу кодування, що може створювати "вузькі місця" при обробці великих потоків даних у реальному часі [17].

1.6. Особливості компресії відеопотоку для задач комп'ютерного зору та машинної ідентифікації об'єктів

Підготовка відеоданих для аналізу алгоритмами комп'ютерного зору вимагає докорінно іншого підходу, ніж підготовка відео для перегляду людиною. Традиційні кодеки базуються на психовізуальній моделі сприйняття, видаляючи інформацію, яку людське око не помічає або вважає менш важливою. Проте для згорткових нейронних мереж, які використовуються для детекції та класифікації об'єктів, саме ці дрібні деталі, текстури та чіткість граней можуть мати вирішальне значення.

Фундаментальною проблемою використання сильно стисненого відео в системах автоматизованої ідентифікації є поява специфічних артефактів. Блочність зображення, "дзвін" навколо контрастних об'єктів та розмиття руху можуть інтерпретуватися нейромережею як хибні ознаки. Це призводить до зниження точності детекції та появи помилок першого і другого роду, це видно на рисунку 1.8.



Рис. 1.8. Порівняння методів компресії

Тому для вхідного потоку, що подається на модуль аналізу, неприпустиме використання агресивних методів компресії, орієнтованих на низький бітрейт.

В контексті розробки системи сортування доцільно розглядати концепцію Video Coding for Machines. Цей підхід передбачає збереження семантичної цілісності

сцени замість візуальної привабливості. Одним із ефективних методів оптимізації потоку є використання адаптивного кодування на основі зон інтересу (ROI)[11,12]. Суть методу полягає в тому, що кадр попередньо аналізується спрощеним алгоритмом для виявлення ділянок з рухом або потенційними об'єктами. Ці ділянки кодуються з мінімальним коефіцієнтом квантування, зберігаючи максимальну деталізацію, тоді як статичний фон піддається сильному стисненню. Такий підхід дозволяє суттєво зменшити навантаження на канал передачі даних всередині системи, не знижуючи при цьому точність роботи класифікатора.

Іншим важливим аспектом є часова надлишковість відеоряду. Для задач сортування медіаконтенту часто немає потреби аналізувати кожен кадр відеопотоку, що надходить зі швидкістю 30 або 60 кадрів на секунду. Ефективною стратегією є вибіркове декодування лише ключових кадрів (I-frames), які містять повну інформацію про зображення і не залежать від попередніх кадрів. Це дозволяє реалізувати дискретний аналіз сцени, зменшуючи обчислювальне навантаження на графічні процесори в десятки разів. Система може аналізувати один-два ключових кадри на секунду, формуючи вектор ознак для класифікації всієї сцени, що є достатнім для точного сортування матеріалу за категоріями.

Також перспективним напрямком є використання векторів руху, які вже обчислені кодеком на етапі стиснення. Замість повного декодування пікселів, система може аналізувати ці вектори для швидкого визначення динаміки в кадрі або типу дії. Це дозволяє реалізувати попередню фільтрацію контенту ще до етапу глибокого навчання, відсіюючи статичні сцени або матеріал, що не містить цільових об'єктів, що значно підвищує загальну швидкодію програмно-апаратного комплексу [12].

1.7. Висновки до розділу 1

У першому розділі дипломної роботи проведено теоретичний аналіз методів компресії відеоданих та принципів роботи систем комп'ютерного зору.

Встановлено, що для довготривалого зберігання відсортованого контенту найбільш ефективними є стандарти H.265 (HEVC) та AV1. Вони забезпечують

зменшення бітрейту на 30–50% порівняно з H.264 при збереженні ідентичної візуальної якості, що є критичним для економії дискового простору.

Визначено фундаментальну відмінність між компресією для людини та для машини. Доведено, що агресивне стиснення та згладжування артефактів знижують точність роботи згорткових нейронних мереж, оскільки вони втрачають високочастотні деталі, такі як текстури та грані.

Обґрунтовано доцільність використання концепції VCM. Для підвищення швидкодії системи запропоновано використовувати методи вилучення ключових кадрів та кодування зон інтересу (ROI), що дозволяє аналізувати контент без необхідності повного декодування всього відеопотоку.

Розглянуто архітектуру згорткових нейронних мереж як основного інструменту ідентифікації. Підтверджено, що саме CNN здатні формувати ієрархічні ознаки зображення, необхідні для класифікації об'єктів в умовах змінної освітленості та ракурсів.

Отримані результати формують теоретичне підґрунтя для проектування архітектури програмного засобу та вибору конкретних інструментів реалізації, що розглядаються у другому розділі.

РОЗДІЛ 2

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО
СОРТУВАННЯ МЕДІАКОНТЕНТУ

2.1. Загальна архітектура системи

Розроблюваний програмний засіб реалізується за принципом конвеєрної архітектури, оскільки послідовна обробка великих обсягів відеоданих вимагає чіткого розмежування функціональних етапів. Основне призначення системи полягає в автоматизованій ідентифікації семантичного змісту відеофайлів та їх подальшому сортуванні у задану користувачем структуру архіву як видно на рисунку 2.1.



Рис. 2.1. Архітектура програми

Архітектура програми складається з чотирьох основних функціональних модулів, які забезпечують виконання повного циклу від завантаження файлу до його архівування:

- Модуль взаємодії з користувачем відповідає за прийом вхідних даних та налаштування правил сортування;

- Модуль попередньої обробки керує декодуванням, вилученням ключових кадрів та їх оптимізацією для нейронної мережі;
- Модуль аналізу та ідентифікації це ядро системи, яке відповідає за виконання алгоритмів комп'ютерного зору та генерацію семантичних ознак;
- Модуль логіки та архівування виконує фінальну класифікацію контенту на основі ознак та переміщення файлу у цільову папку.

Робота системи є строго послідовною та ітеративною для кожного вхідного відеофайлу, що забезпечує надійність обробки та легкість трасування помилок під час розробки.

Таблиця 2.1

Функціональні вимоги до системи

Функціональна вимога	Опис реалізації
Керування вхідними даними	Можливість обробки одиночних файлів або цілих директорій в пакетному режимі.
Ефективна попередня обробка	Реалізація вилучення лише ключових кадрів для аналізу, забезпечення масштабування та нормалізації зображення під вхідні вимоги моделі YOLO.
Ідентифікація об'єктів	Детекція та класифікація об'єктів у реальному часі за допомогою моделі CV з можливістю налаштування порогу впевненості.
Настроювана логіка сортування	Можливість створення користувачем гнучких правил сортування на основі комбінацій знайдених об'єктів та їх кількості.
Оптимізоване архівування	Опціональне транскодування фінального файлу в ресурсоефективний кодек H.265 перед збереженням.
Збереження метаданих	Автоматична генерація файлів-звітів у форматі JSON із усіма ідентифікованими ознаками, що забезпечує подальший пошук і фільтрацію.

2.2. Приклад використання та функціональні вимоги

Користувач через інтерфейс задає директорію-джерело та директорію-архів. У цьому ж модулі визначаються правила сортування, які є ключовою частиною логіки. Наприклад, якщо знайдено об'єкти "людина" і "сніг", то відео потрапляє в папку "Зимові_події".

Модуль відеопроцесингу отримує відеофайл і, використовуючи бібліотеки, такі як FFmpeg/OpenCV[17,18], виконує вибіркове декодування. На цьому етапі реалізується концепція компресії для машин, коли вилучаються лише ключові кадри (I-frames) або кадри з фіксованим інтервалом. Це значно знижує обчислювальне навантаження, адже аналізу підлягають лише репрезентативні знімки сцени. Вилучені кадри нормалізуються до необхідного вхідного розміру нейромережі.

Підготовлені кадри надходять до Модуля аналізу. Тут відбувається основний процес інференсу за допомогою обраної моделі. Нейронна мережа швидко обробляє кожен кадр, повертаючи список ідентифікованих об'єктів, їхні обмежувальні рамки та відповідні показники впевненості. Ці результати агрегуються у загальний вектор семантичних ознак для всього відеофайлу. Наприклад, якщо у 80% кадрів була виявлена "вода" та "корабель", ці ознаки отримують високий кумулятивний бал.

Модуль логіки та архівування бере згенерований вектор ознак та застосовує до нього встановлені правила сортування. На основі цих правил визначається фінальна категорія відео. Після визначення категорії виконується:

1. Якщо вхідний файл має низьку компресію, програма стискає його до стандарту H.265, зберігаючи якість, але мінімізуючи розмір;
2. Створюється супровідний файл, наприклад, JSON, що містить список усіх знайдених тегів та фінальну категорію;
3. Файл переміщується у відповідну підпапку архіву.

2.3. Обґрунтування вибору моделі комп'ютерного зору для підсистеми ідентифікації відеоконтенту

В умовах сучасного інформаційного простору, що характеризується експоненційним зростанням обсягів мультимедійних даних, задача автоматизованої ідентифікації та сортування відеоконтенту набуває критичного значення. Для дипломного дослідження, метою якого є розробка ефективних методів та програмно-апаратних засобів для роботи з медіаконтентом, вибір базової архітектури нейронної мережі є фундаментом, що визначає кінцеву ефективність усієї системи. На відміну від обробки статичних зображень, робота з відеопотоками висуває специфічні та жорсткі вимоги до алгоритмів комп'ютерного зору. Відеодані характеризуються високою часовою надлишковістю, наявністю ефектів розмиття при русі, зміною освітлення та складною композицією сцени, що вимагає від детектора не лише високої точності локалізації об'єктів, але й стабільності детекції у часі.

Ключовим викликом при проектуванні системи сортування є досягнення балансу між трьома взаємозалежними метриками:

- точністю, яка вимірюється через середню точність;
- швидкодією, що визначає можливість роботи в реальному часі;
- та ресурсоефективністю, яка впливає на вартість масштабування системи.

Традиційні методи, що базуються на ручному виділенні ознак або класичних архітектурах машинного навчання, вичерпали свій потенціал у цій сфері, поступившись місцем згортковим нейронним мережам та трансформерним архітектурам.

У даному частині проводиться глибинний аналіз сучасних архітектур детекції об'єктів, розглядається їх математичний апарат, архітектурні особливості та показники продуктивності. Метою аналізу є аргументація вибору сімейства моделей YOLO версій v11 як оптимального рішення для задачі сортування медіаконтенту, базуючись на емпіричних даних, бенчмарках та теоретичних перевагах у швидкодії та масштабованості.

Сучасний ландшафт моделей детекції об'єктів можна класифікувати за двома основними напрямками: архітектури на основі CNN та архітектури на основі

трансформерів. Кожен з цих підходів має унікальні характеристики, що визначають їх придатність для конкретних сценаріїв використання.

Історично важливим етапом розвитку детекції об'єктів стала поява двостадійних детекторів, найяскравішим представником яких є Faster R-CNN[14]. Алгоритмічна суть цього підходу полягає у розділенні процесу детекції на два послідовних етапи, що дозволяє досягти високої точності локалізації, але створює суттєві обмеження для швидкодії.

На першому етапі функціонує мережа пропозиції регіонів. Вона сканує карту ознак, отриману від магістральної мережі, та генерує набір потенційних регіонів інтересу, де з високою ймовірністю знаходиться об'єкт. Математично це реалізується через ковзне вікно, що класифікує кожен "якір" як передній план або фон.

На другому етапі виділені регіони проходять через процедуру RoI Pooling, яка приводить їх до фіксованого розміру, після чого вони подаються на вхід повнозв'язним шарам для остаточної класифікації та регресії координат обмежувальної рамки.

Безперечною перевагою Faster R-CNN є висока точність, особливо при роботі з малими об'єктами та сценами з високою щільністю перекриття об'єктів. Це досягається завдяки детальному аналізу кожного запропонованого регіону. Однак, для задачі сортування відеопотоків цей архітектурний підхід має критичні недоліки.

По-перше, двостадійна природа алгоритму створює обчислювальне "вузьке місце". Необхідність обробки сотень пропозицій регіонів для кожного кадру призводить до значної затримки. На сучасному обладнанні, наприклад, NVIDIA T4 швидкість інференсу Faster R-CNN рідко перевищує 15-20 кадрів на секунду, що є недостатнім для обробки відео в реальному часі або для високошвидкісного сортування архівів.

По-друге, ресурсомісткість моделі є надмірною. Зберігання проміжних карт ознак для кожного регіону вимагає значного обсягу відеопам'яті, що ускладнює розгортання системи на периферійних пристроях або обмежує кількість паралельних потоків, які може обробляти один сервер. Враховуючи вимоги де важлива економічна ефективність, використання Faster R-CNN є недоцільним.

Альтернативою двостадійним системам стали одностадійні детектори, які розглядають задачу детекції як єдину задачу регресії, прогножуючи класи та координати об'єктів безпосередньо з повної карти ознак за один прохід мережі.

SSD (Single Shot MultiBox Detector) Архітектура, описана на рисунку 2.2, стала піонером використання пірамідальних карт ознак для детекції об'єктів різного масштабу. Використовуючи карти ознак з різних шарів мережі, SSD здатна детектувати як великі об'єкти, так і малі об'єкти. Це забезпечило значний приріст швидкості порівняно з Faster R-CNN. Однак, ранні версії SSD страждали від дисбалансу класів, де переважаючий фон пригнічував градієнти від об'єктів під час навчання.

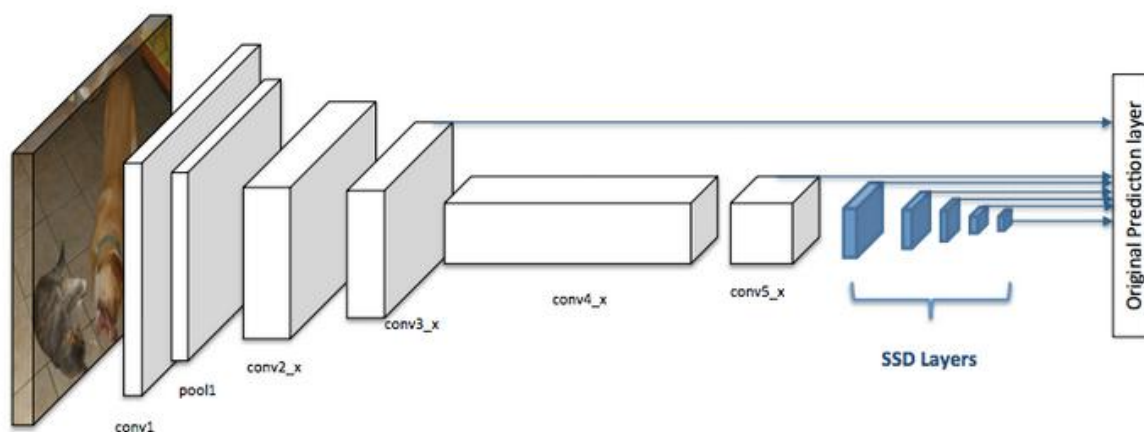


Рис. 2.2. Схема архітектури SSD моделі

Проривом у одностадійних детекторах стала архітектура RetinaNet, яка ввела поняття Focal Loss — модифікованої функції втрат перехресної ентропії. Математично вона визначається формулою (2.1) [16].

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (2.1)$$

де p_t — ймовірність класу, а $(1 - p_t)^\gamma$ — модулюючий фактор. Ця формула дозволяє динамічно зменшувати вагу "легких" прикладів і фокусувати навчання на

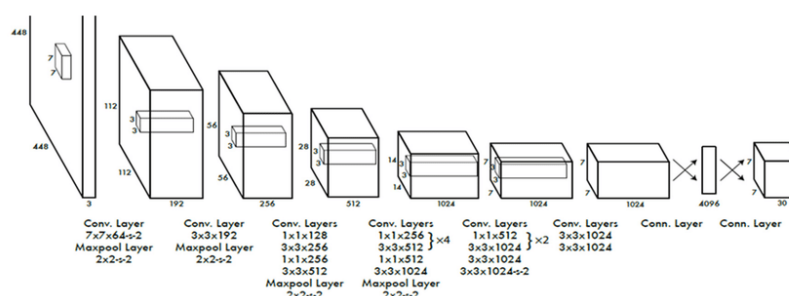
"складних" прикладах. Це дозволило одностадійним детекторам досягти точності, порівнянної з двостадійними, зберігаючи при цьому високу швидкість.

З появою механізму уваги у обробці природної мови, ці підходи почали адаптуватися і для комп'ютерного зору. DETR запропонував революційний підхід, що розглядає детекцію як задачу прогнозування множини.

Основна інновація трансформерних детекторів полягає у відмові від концепції якорів та пост-процесингу NMS. Замість цього, мережа використовує механізм двочастинного співставлення та Угорський алгоритм для розрахунку функції втрат, що забезпечує унікальне призначення кожного прогнозу істинному об'єкту.

Еволюція DETR призвела до появи RT-DETR, першого трансформерного детектора, здатного працювати в реальному часі. RT-DETR використовує ефективний гібридний енкодер, який обробляє мультимасштабні ознаки, відокремлюючи внутрішньомасштабну взаємодію від крос-масштабного злиття. Незважаючи на високу точність та відсутність NMS, трансформерні моделі мають суттєвий недолік у контексті ресурсомісткості. Механізм self-attention має квадратичну складність відносно кількості токенів, що призводить до значного споживання пам'яті та високого навантаження на обчислювальні блоки, особливо при високій роздільній здатності зображення.

Сімейство моделей YOLO де-факто стало стандартом індустрії для задач реального часу завдяки постійній інтеграції передових дослідницьких рішень у практичну архітектуру. Для реалізації задачі розглядаються три останні знакові версії: YOLOv8, YOLOv10 та YOLOv11, кожна з яких пропонує унікальні переваги для задачі ідентифікації медіаконтенту[20]. Їх архітектура показан на рисунку 2.2.



Випущена на початку 2023 року компанією Ultralytics, модель YOLOv8 ознаменувала перехід до повністю безякірної архітектури, що спростило процес навчання та покращило узагальнюючу здатність моделі.

Архітектурні особливості:

- Backbone основою якої слугує модифікована архітектура CSPDarknet53. Ключовою зміною стало введення блоку C2f, який замінив блок C3 з попередніх версій. Блок C2f розроблений на основі ідей ELAN і забезпечує багатший градієнтний потік під час зворотного поширення помилки. Це досягається шляхом додавання більшої кількості розгалужень та злиттів тензорів ознак, що дозволяє моделі навчатися більш складним репрезентаціям без суттєвого збільшення обчислювальної вартості[21];

- Decoupled Head, на відміну від попередніх версій, де класифікація та регресія виконувались у спільних шарах, YOLOv8 розділяє ці задачі. Це логічно обґрунтовано тим, що ознаки, необхідні для визначення класу об'єкта, відрізняються від ознак, необхідних для точного визначення його меж. Розділення голів прискорює конвергенцію моделі та підвищує точність[21];

- Task Aligned Assigner механізм динамічного призначення зразків базується на оцінці якості вирівнювання задачі. Позитивні зразки обираються на основі зваженої комбінації оцінки класифікації та IoU, що гарантує навчання на найбільш якісних прикладах.

YOLOv10, представлена дослідниками з університету Цінхуа у травні 2024 року. Він вирішує фундаментальну проблему всіх попередніх детекторів YOLO — залежність від пост-процесингу NMS.

Проблема NMS в тому, що традиційні детектори генерують безліч дубльованих рамок навколо одного об'єкта. NMS — це евристичний алгоритм, який фільтрує ці рамки, залишаючи лише одну з найвищою довірою. Однак, NMS є послідовною операцією, швидкість якої залежить від кількості детекцій, що створює нестабільну затримку у системі.

Рішення YOLOv10 в Consistent Dual Assignments. Архітектура YOLOv10 вводить інноваційну стратегію навчання з подвійним призначенням міток:

- Голова "Один-до-багатьох" (One-to-many Head). Працює як у класичних YOLO, призначаючи кілька прогнозів одному об'єкту. Це забезпечує багатий наглядний сигнал (rich supervision) для ефективного навчання магістралі.
- Голова "Один-до-одного" (One-to-one Head). Працює за принципом DETR, призначаючи лише один прогноз на один істинний об'єкт.

Під час навчання обидві голови оптимізуються спільно, використовуючи узгоджену метрику відповідності за формулою 2.2. Під час експлуатації використовується лише голова "Один-до-одного", що дозволяє отримувати фінальні результати без NMS. Це знижує затримку на 30-50% порівняно з YOLOv8 [22].

$$m(\alpha, \beta) = s \cdot p^\alpha \cdot IoU(b, b^{gt})^\beta \quad (2.2)$$

де s — індикатор просторового пріору, p — ймовірність класу, b та b^{gt} — прогнозована та істинна рамки. YOLOv10 використовує ідентичні параметри α та β для обох голів, що гарантує узгодженість їхніх прогнозів.

Випущена у жовтні 2024 року, YOLOv11 від Ultralytics представляє собою подальшу оптимізацію архітектури з метою підвищення ефективності параметрів та точності на складних сценах

Ключова інновація версії в веддені двох блоків.

Блок C3k2, Замінює блок C2f на C3k2. Цей блок використовує згортки з меншим розміром ядра, що дозволяє зменшити кількість параметрів та обчислювальних операцій при збереженні того ж рецептивного поля. Це робить модель більш компактною та швидкою, особливо на GPU [21,22].

Блок C2PSA. Найважливішим нововведенням для аналізу медіаконтенту є введення механізму просторової уваги. Блок C2PSA додається після модуля SPPF і дозволяє моделі фокусуватися на інформативних ділянках зображення, ігноруючи фоновий шум.

YOLOv11m має на 22% менше параметрів, ніж YOLOv8m, при цьому демонструючи вищу точність на COCO датасеті. Це дозволяє ефективніше використовувати кеш процесора та відеопам'ять.

Для обґрунтованого вибору моделі необхідно провести кількісне порівняння ключових показників продуктивності. У Таблиці 2.2 наведено зведені дані бенчмаркінгу на стандартному датасеті COCO Val 2017, отримані на апаратному забезпеченні NVIDIA T4 (з використанням TensorRT), що є репрезентативним для серверних рішень середнього рівня [23].

Таблиця 2.2

Порівняння характеристик моделей CV

Модель	Розмір входу	mAP (50-95)	Параметри (M)	FLOPs (B)	Latency T4 TensorRT (ms)	FPS	Особливості
YOLOv8n	640	37.3	3.2	8.7	2.1	~470	Базова anchor-free модель
YOLOv8m	640	50.2	25.9	78.9	6.5	~150	Стабільна середня модель
YOLOv10n	640	39.5	2.3	6.7	1.84	~540	NMS-free, найменша затримка
YOLOv10m	640	51.3	15.4	59.1	4.74	~210	Висока ефективність параметрів
YOLOv11n	640	39.5	2.6	6.5	1.5	~660	Найвища швидкість, C3k2 блоки
YOLOv11m	640	51.5	20.1	68.0	4.7	~212	Найкращий баланс mAP/FPS
RT-DETR-L	640	53.4	42	136	9.76	~100	Висока точність, але важка навантаження
Faster R-CNN	-	~41.0	40+	200+	40+	~25	Застаріла для Real-Time

Як видно з таблиці, моделі YOLOv11 демонструють суттєву перевагу у швидкодії. YOLOv11m здатна обробляти відео зі швидкістю понад 200 FPS на GPU T4, що у 8 разів швидше за Faster R-CNN та вдвічі швидше за RT-DETR-L. Це означає, що на одному фізичному сервері з однією картою T4 можна одночасно обробляти до 6-7 потоків відео 1080p (30 FPS) у реальному часі. Використання RT-DETR дозволило б обробляти лише 2-3 потоки через вищі вимоги до обчислень та пам'яті. YOLOv10n та YOLOv11n показують феноменальні результати для Edge-пристроїв (понад 500 FPS), що відкриває можливості для реалізації системи сортування безпосередньо на камерах або мобільних платформах.

Моделі YOLO мають найкращу підтримку в екосистемі розгортання:

- Завдяки простій архітектурі (тільки згортки, відсутність складних механізмів уваги трансформерів), YOLOv11 легко конвертується в оптимізований формат TensorRT Engine, забезпечуючи максимальну утилізацію ядер CUDA.
- Підтримка стандарту ONNX дозволяє запускати ці моделі на CPU з прийнятною швидкістю, що робить систему гнучкою та незалежною від наявності дискретної GPU.

Трансформерні моделі страждають від квадратичної залежності споживання пам'яті від розміру входу. Для обробки 4K відео це стає критичним бар'єром. CNN-архітектура YOLO масштабується лінійно, що дозволяє працювати з високою роздільною здатністю, необхідною для ідентифікації дрібних деталей у медіаконтенті.

2.4. Вибір мови програмування та середовища розробки

Для реалізації програмного комплексу автоматизованого сортування медіаконтенту обрано мову програмування Python версії 3.10 [24]. Цей вибір зумовлений домінуванням Python у сфері машинного навчання та комп'ютерного зору. На відміну від компільованих мов (C++), Python забезпечує високу швидкість розробки та прототипування завдяки простому синтаксису та динамічній типізації. Ключовим фактором стала наявність розвиненої екосистеми бібліотек, що дозволяє

інтегрувати складні математичні алгоритми та нейронні мережі за допомогою високорівневих API, не витрачаючи час на низькорівневе керування пам'яттю. У якості інтегрованого середовища розробки (IDE) обрано Visual Studio Code.

Програмна реалізація базується на використанні наступних спеціалізованих бібліотек:

- OpenCV основний інструмент для роботи з відеопотоком. У даній роботі бібліотека cv2 використовується для модуля попередньої обробки. Вона реалізує зчитування відеофайлів, декодування кадрів, зміни їх роздільної здатності та конвертації колірних просторів з BGR у RGB перед подачею на вхід нейромережі.

- Ultralytics YOLO. Для реалізації модуля детекції обрано фреймворк Ultralytics, який надає зручний інтерфейс для роботи з архітектурою YOLO. Це дозволяє використовувати попередньо навчені ваги на датасеті COCO, що забезпечує розпізнавання 80 класів об'єктів "з коробки" з високою продуктивністю на CPU та GPU [21].

- PyTorch: Виступає як бекенд для роботи нейронної мережі. Бібліотека забезпечує ефективне виконання тензорних обчислень, необхідних для роботи згорткових шарів моделі.

- NumPy використовується для високоефективних математичних операцій з багатовимірними масивами, у вигляді яких зображення передаються між OpenCV та нейронною мережею.

2.5. Опис алгоритму роботи програми

Алгоритм функціонування розробленої системи можна представити у вигляді послідовності кроків, що виконуються для кожного відеофайлу у вхідній директорії.

Як видно на рисунку 2.3, робота розпочинається з ініціалізації, де програма завантажує ваги нейронної мережі YOLO в оперативну пам'ять та зчитує конфігураційний файл, який містить шляхи до папок та правила класифікації.

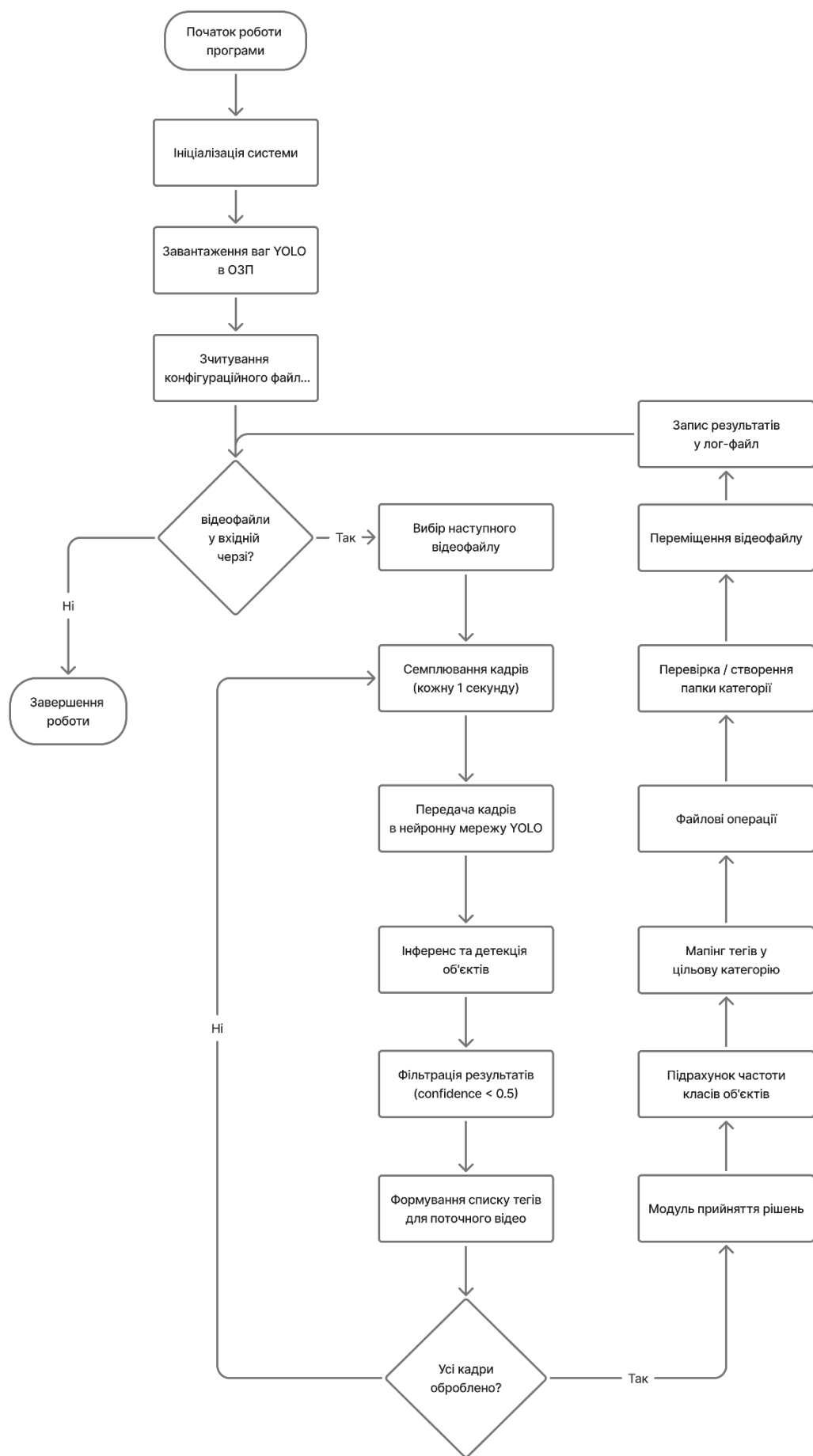


рис. 2.3. Алгоритм роботи програми

На етапі циклічної обробки, система сканує вхідну директорію. Для кожного виявленого відеофайлу запускається процес семплювання кадрів. Замість декодування всього потоку, алгоритм витягує кадри з фіксованим кроком, наприклад, кожну 1 секунду. Це дозволяє отримати репрезентативну вибірку сцени без надлишкових обчислень.

Отримані кадри передаються на вхід нейронної мережі. Для кожного кадру модель повертає список детектованих об'єктів та ступінь впевненості. Програма фільтрує результати, відкидаючи об'єкти з низькою впевненістю (менше 0.5), та формує тимчасовий список знайдених тегів для поточного файлу.

Після обробки всіх кадрів відеофайлу активується модуль прийняття рішень. Алгоритм підраховує частоту появи кожного класу об'єктів. На основі розробленої логіки пріоритетів та мапінгу, сукупність тегів перетворюється на єдину цільову категорію.

Завершальним етапом є виконання файлових операцій. Програма перевіряє наявність папки з назвою категорії (створює її, якщо вона відсутня) та переміщує туди відеофайл. Паралельно генерується лог-файл, куди записується інформація про результати сортування. Після цього цикл повторюється для наступного файлу в черзі.

2.6. Розробка алгоритму прийняття рішень та логіки класифікації

Для запобігання надмірній фрагментації файлового архіву та виникненню неконтрольованої кількості директорій, у системі реалізовано модуль прийняття рішень на основі фіксованої таксономії категорій.

На відміну від "сирого" виходу нейронної мережі, який може містити до 80 різних класів об'єктів згідно датасету MS COCO, цільова структура архіву обмежується заздалегідь визначеним набором категорій, що задається адміністратором системи у конфігураційному файлі.

Процес визначення фінальної категорії базується на трьох етапах:

- Семантична агрегація. Вхідні класи об'єктів, детектовані моделлю комп'ютерного зору, групуються у узагальнені кластери. Реалізація цього етапу базується на матриці відповідності і агрегуються у категорію.
- Статистична фільтрація. Присутність об'єкта в одиничних кадрах може бути статистичною похибкою або шумом детекції. Для присвоєння категорії вводиться поріг частоти появи. Об'єкт враховується при класифікації лише тоді, коли він присутній у більш ніж ключових кадрах відеопотоку.
- Вирішення конфліктів пріоритетів. У випадках, коли відеоряд містить об'єкти з різних агрегованих груп, система використовує ієрархію пріоритетів. Відеофайл класифікується за категорією з найвищим пріоритетом серед виявлених. У разі, якщо жоден із об'єктів не відповідає правилам фільтрації, файл переміщується до спеціальної директорії "Unsorted" для ручної перевірки, що гарантує цілісність структури архіву.

2.7. Висновки до розділу 2

У другому розділі дипломної роботи проведено обґрунтування вибору методів та програмних засобів для реалізації системи автоматизованої ідентифікації та сортування медіаконтенту.

Розроблено модульну конвеєрну архітектуру системи, яка забезпечує послідовну обробку даних на етапах попередньої підготовки, нейромережевого аналізу та фінального архівування. Такий підхід гарантує масштабованість програми та можливість обробки великих масивів відеоданих.

Як основний інструментарій обрано мову програмування Python у зв'язці з бібліотеками OpenCV для відеопроесування та PyTorch для обчислень. Це забезпечує гнучкість розробки та високу сумісність компонентів.

Ключовим елементом системи визначено нейронну мережу YOLO11. Її використання обґрунтовано найкращим співвідношенням швидкості роботи та точності детекції, що дозволяє виконувати аналіз контенту в режимі реального часу без надмірних вимог до апаратного забезпечення.

Вирішено проблему семантичної неоднозначності шляхом розробки логічного модуля, який перетворює "сирі" класи об'єктів у фіксовану структуру папок архіву. Запроваджено систему пріоритетів та частотних порогів, що мінімізує помилки сортування.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Розгортання середовища розробки та його налаштування

Для реалізації розробленого алгоритму сортування медіаконтенту було обрано сучасний набір технологій, що забезпечує високу швидкодію та простоту масштабування. Цей підрозділ описує процес підготовки робочого середовища, встановлення необхідних залежностей та обґрунтування вибору конкретних програмних бібліотек.

В якості основної мови програмування використовується Python версії 3.12.10. Ця версія є однією з найстабільніших на момент розробки та забезпечує покращену продуктивність та вдосконалену роботу з типізацією даних [24].

В якості інтегрованого середовища розробки обрано Visual Studio Code. Його вибір зумовлений такими факторами:

- Легковажність та модульність, можливість встановлення розширень для Python, що забезпечують інтелектуальне автодоповнення коду та статичний аналіз помилок;
- Інтегрований термінал, що дозволяє керувати віртуальними середовищами та пакетами `pip` безпосередньо у вікні редактора;
- Підтримка контролю версій. Вбудована робота з Git, що є стандартом для сучасної розробки ПЗ.

Для запобігання конфліктам версій бібліотек між різними проектами було використано механізм віртуальних середовищ `venv`. Це дозволяє створити ізольований контейнер для проекту, де зберігаються лише необхідні для його роботи пакети. Команда зображена на рисунку 3.1.

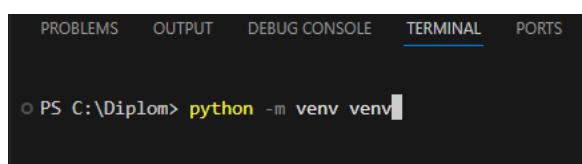


рис. 3.1. Команда ініціалізації `venv`

Команда створила необхідні файли, як на рисунку 3.2 в директорії проекту і дала можливість запустити `venv`.



рис. 3.2. Файли `venv`

Активация середовища дозволила встановити специфічні версії бібліотек, не засмічуючи глобальний інтерпретатор Python операційної системи. Нижче наведено перелік ключових компонентів, що складають ядро розробленої системи і описана в таблиці 3.1.

Таблиця 3.1

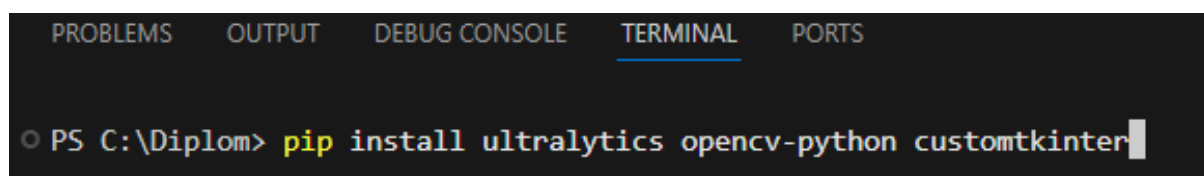
Використані бібліотеки

Бібліотека	Призначення та роль у програмі
Ultralytics (YOLO11)	Ядро системи комп'ютерного зору. Бібліотека забезпечує завантаження ваг нейронної мережі YOLO11n, попередню обробку зображень та виконання детекції об'єктів. Вона повертає координати знайдених об'єктів та їх класові мітки.
OpenCV (cv2)	Обробка відеопотоку. Використовується для зчитування відеофайлів різних форматів (.mp4, .avi), розбиття відео на окремі кадри та отримання технічних метаданих (FPS, роздільна здатність, загальна кількість кадрів).

Продовження таблиці 3.1

CustomTkinter	Графічний інтерфейс користувача. Сучасна обгортка над стандартною бібліотекою Tkinter. Забезпечує створення естетичного інтерфейсу програми [25].
NumPy	Математичні обчислення. Використовується як допоміжна бібліотека для роботи з матрицями зображень, які передаються між OpenCV та нейронною мережею.
OS i Shutil	Файлові операції. Стандартні модулі Python, що відповідають за навігацію по файловій системі, створення директорій категорій (наприклад, "Transport", "Animals") та фізичне переміщення оброблених файлів.
Threading	Багатопотоковість. Забезпечує виконання ресурсоемних задач аналізу відео в окремому потоці, що запобігає "зависанню" графічного інтерфейсу під час роботи нейромережі.

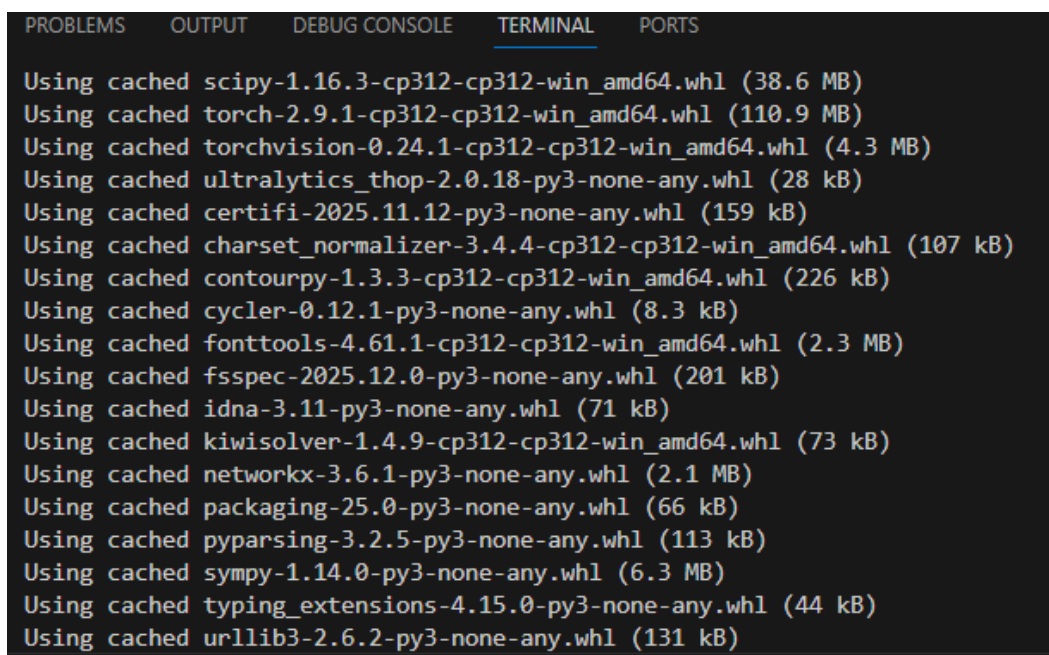
Встановлення бібліотек виконувалося за допомогою менеджера пакетів `pip`. Команда яка ініціалізує встановлення необхідних компонентів описана на рисунку 3.3



The image shows a terminal window with a dark background. At the top, there are tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected and underlined), and PORTS. Below the tabs, the command prompt shows 'PS C:\Diplom>' followed by the command 'pip install ultralytics opencv-python customtkinter' in a light blue color. A white cursor is at the end of the command.

рис. 3.3. команда для встановлення бібліотек

Це привело до початку встановлення бібліотек і наступному консольному виводу на рисунку 3.4 який показує список встановлених бібліотек і засобів для реалізації програми.



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Using cached scipy-1.16.3-cp312-cp312-win_amd64.whl (38.6 MB)
Using cached torch-2.9.1-cp312-cp312-win_amd64.whl (110.9 MB)
Using cached torchvision-0.24.1-cp312-cp312-win_amd64.whl (4.3 MB)
Using cached ultralytics_thop-2.0.18-py3-none-any.whl (28 kB)
Using cached certifi-2025.11.12-py3-none-any.whl (159 kB)
Using cached charset_normalizer-3.4.4-cp312-cp312-win_amd64.whl (107 kB)
Using cached contourpy-1.3.3-cp312-cp312-win_amd64.whl (226 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Using cached fonttools-4.61.1-cp312-cp312-win_amd64.whl (2.3 MB)
Using cached fsspec-2025.12.0-py3-none-any.whl (201 kB)
Using cached idna-3.11-py3-none-any.whl (71 kB)
Using cached kiwisolver-1.4.9-cp312-cp312-win_amd64.whl (73 kB)
Using cached networkx-3.6.1-py3-none-any.whl (2.1 MB)
Using cached packaging-25.0-py3-none-any.whl (66 kB)
Using cached pyparsing-3.2.5-py3-none-any.whl (113 kB)
Using cached sympy-1.14.0-py3-none-any.whl (6.3 MB)
Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Using cached urllib3-2.6.2-py3-none-any.whl (131 kB)

```

рис. 3.4. Файли venv

Таким чином було підготовлено середовище розробки, встановлено бібліотеки і налаштовано віртуалізацію для подальшої реалізації програми.

3.2. Розгортання моделі YOLO11 та перевірка її працездатності

Після налаштування програмного середовища наступним критичним етапом є інтеграція та тестування ядра системи, а саме нейронної мережі YOLO11. Взаємодія з моделлю здійснюється через високорівневий API бібліотеки ultralytics, який дозволяє виконувати складні операції завантаження ваг, налаштування параметрів інференсу та обробки результатів за допомогою лаконічних команд мови Python.

Життєвий цикл моделі в програмі ідентифікації відео складається з трьох етапів.

Ініціалізація та завантаження ваг. Для дипломного проекту було обрано конфігурацію YOLO11 Nano (yololln.pt). Це найлегша версія архітектури, яка забезпечує найвищу швидкість обробки кадрів, що є пріоритетом для сортування великих відеоархівів на персональних комп'ютерах.

Основний метод “model()” приймає зображення або відеопотік та повертає об'єкт “Results”. Цей метод підтримує ряд параметрів для налаштування чутливості системи:

- Conf (поріг впевненості), за замовчуванням становить 0.25. Для зменшення кількості хибних спрацювань у роботі встановлено значення 0.5.

- Verbose відповідає за керування виводом технічної інформації у консоль.

- Device дає можливість вибору обчислювального пристрою.

Отриманий об'єкт результату містить масиви даних:

- boxes.xxyy. координати рамок об'єктів.

- boxes.conf. Рівень впевненості для кожного об'єкта.

- boxes.cls. Ідентифікатори класів, які необхідно декодувати в текстові назви, наприклад, 0 це 'person'.

Для перевірки коректності встановлення бібліотек та працездатності моделі було проведено контрольний експеримент. В якості тестових даних використано стандартне зображення "bus.jpg", яке містить об'єкти класів "person" та "bus" [21,26]. Було розроблено та виконано наступний тестовий код зображений на рисунку 3.5.

```
YoloTest.py > ...
1  from ultralytics import YOLO
2
3  # 1. Ініціалізація моделі
4  print("Завантаження моделі YOLO11...")
5  model = YOLO("yolo11n.pt")
6
7  # 2. Тестовий інференс на віддаленому зображенні
8  # Використовується зображення з репозиторію Ultralytics
9  results = model("https://ultralytics.com/images/bus.jpg")
10
11 # 3. Візуалізація та вивід результатів
12 for result in results:
13     result.show() # Відкриття вікна з результатами детекції
14
15     # Вивід знайдених класів у консоль
16     print("Результати детекції:")
17     for cls_id, count in result.verbose().items():
18         print(f" - Клас: {cls_id}, Кількість: {count}")
```

рис. 3.5. Тестовий код YoloTest.py

В результаті виконання скрипту модель успішно завантажила ваги yolo11n.pt з хмарного сховища та виконала детекцію об'єктів. У консоль було виведено статистику, яка підтвердила ідентифікацію об'єктів. Візуалізація результату на рисунку 3.6 демонструє, що нейронна мережа коректно побудувала обмежувальні рамки навколо людей та автобуса, правильно класифікувавши їх навіть в умовах часткового перекриття об'єктів.

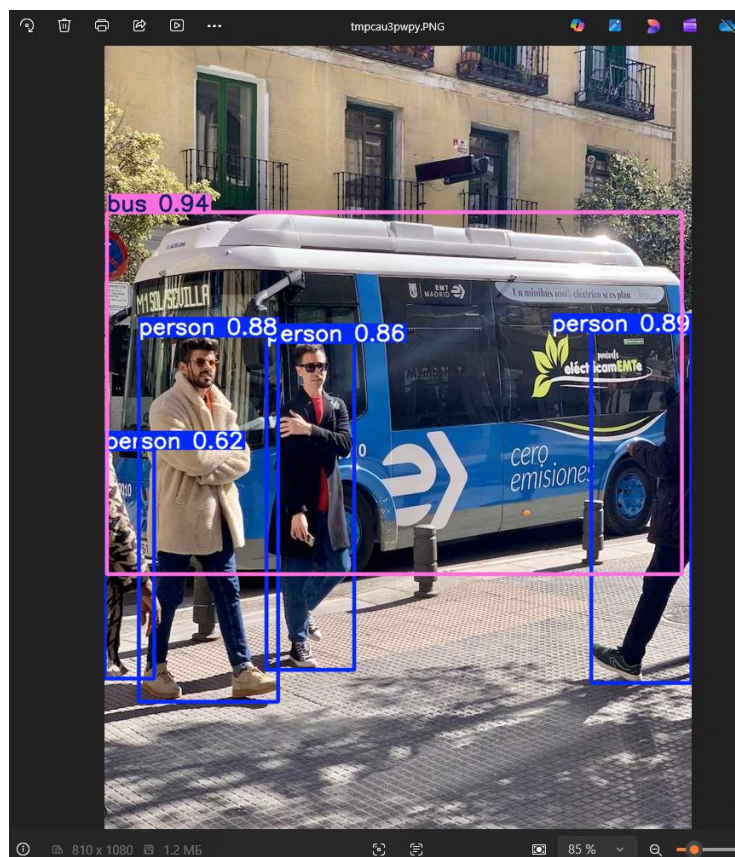


рис. 3.6. Візуалізація результату роботи моделі

Це демонструє, що середовище розгорнуто коректно, модель функціонує згідно з заявленими характеристиками, а бібліотека Ultralytics успішно інтегрована в проект.

3.3. Програмна реалізація функціоналу ідентифікації відеоконтенту

Цей підрозділ присвячено опису ключових алгоритмів, що відповідають за інтелектуальний аналіз відеоданих. Розроблена підсистема поєднує в собі графічний інтерфейс для налаштування параметрів детекції та бекенд-логіку, яка виконує

покадрову обробку відеопотоку, агрегацію статистичних даних та прийняття рішення про категорію контенту.

Для забезпечення гнучкості системи та можливості проведення порівняльного аналізу, в програмі реалізовано механізм динамічного вибору архітектури нейронної мережі. Користувач має можливість через графічний інтерфейс (ComboBox) обрати одну з доступних моделей (YOLOv8 або YOLO11 різної складності).

Технічна реалізація базується на використанні словника відповідностей (MODELS_MAP), який відокремлює зрозумілий для користувача опис моделі від її системного ідентифікатора. Це дозволяє уникнути помилок при завантаженні файлів та легко розширювати список підтримуваних моделей. Код описано на рисунку 3.7.

```
# Словник відповідностей
MODELS_MAP = {
    "yolo11n (Nano - найшвидша, наш стандарт)": "yolo11n.pt",
    "yolo11s (Small - точніша, але повільніша)": "yolo11s.pt",
    "yolov8n (v8 Nano - для порівняння)": "yolov8n.pt",
    "yolov8s (v8 Small - важча версія v8)": "yolov8s.pt"
}

def process_videos(self):
    # Отримання вибору користувача з GUI
    selected_label = self.selected_model_label.get()

    # отримання імені файлу зі словника
    model_filename = MODELS_MAP.get(selected_label)

    self.log(f"--> [INIT] Завантаження файлу ваг: {model_filename}...")

    try:
        # Ініціалізація об'єкта YOLO з бібліотеки Ultralytics
        model = YOLO(model_filename)
    except Exception as e:
        self.log(f"--> [ERROR] Помилка завантаження моделі: {e}")
    return
```

рис. 3.7. Код вибору моделі аналізу відео

Оскільки обробка кожного кадру відео є надлишковою і ресурсоємною операцією, розроблено алгоритм адаптивного семплювання. Програма зчитує метадані відеофайлу за допомогою бібліотеки OpenCV, визначає частоту кадрів і налаштовує крок обробки (`frame_step`) таким чином, щоб аналізувати лише один кадр на секунду. Це забезпечує баланс між швидкістю роботи та точністю класифікації сцени. У циклі обробки виконується детекція лише для обраних кадрів. Результати фільтруються за порогом впевненості (`conf > 0.5`), щоб виключити хибні спрацювання. Код описано на рисунку 3.8.

```
def analyze_single_video(self, model, video_path):
    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened(): return "Error", {}

    # Отримання FPS для розрахунку кроку
    fps = cap.get(cv2.CAP_PROP_FPS)

    # Оптимізація: обробляємо 1 кадр на секунду
    frame_step = int(fps) if fps > 0 else 30

    detected_objects = []
    frame_count = 0

    while True:
        ret, frame = cap.read()
        if not ret: break # Кінець файлу

        # Перевірка умови кроку
        if frame_count % frame_step == 0:
            # Виконання детекції (verbose=False вимикає зайвий вивід у консоль)
            results = model(frame, verbose=False)

            # Обробка результатів детекції
            for result in results:
                for box in result.bboxes:
                    # Фільтрація за впевненістю (> 50%)
                    if float(box.conf[0]) > 0.5:
                        cls_id = int(box.cls[0])
                        # Збереження текстової назви класу
                        detected_objects.append(model.names[cls_id])

            frame_count += 1
        cap.release()

    # Повертаємо "сирий" список усіх знайдених об'єктів
    return detected_objects
```

рис. 3.8. Код семплювання і аналізу відео

Після завершення сканування файлу програма отримує список усіх детектованих об'єктів (наприклад, ['car', 'car', 'person', 'car']). Для перетворення цих даних у фінальну категорію використовується модуль прийняття рішень, який базується на наперед визначеній таксономії категорій (CATEGORY_MAPPING).

Алгоритм підраховує кількість входжень кожного об'єкта та сумує бали для відповідних категорій. Категорія, що набрала найбільшу кількість "голосів", визначається як цільова. Якщо жоден об'єкт не знайдено або вони не відповідають жодній категорії, присвоюється статус "Unsorted". Код описано на рисунку 3.9.

```
# Таксономія категорій
CATEGORY_MAPPING = {
    "Transport": ["car", "bus", "truck", "airplane", "boat"],
    "Animals":   ["bird", "cat", "dog", "horse", "sheep"],
    "People":    ["person"],
    "Sports":    ["sports ball", "skateboard", "surfboard", "skis"],
    "Tech":      ["laptop", "mouse", "keyboard", "cell phone"]
}

# Підрахунок статистики знайдених об'єктів
object_counts = Counter(detected_objects)
category_scores = Counter()

# Агрегація балів по категоріях
for obj, count in object_counts.items():
    found = False
    for category, items in CATEGORY_MAPPING.items():
        if obj in items:
            category_scores[category] += count
            found = True
            break
    # Якщо об'єкт невідомий системі
    if not found:
        category_scores["Unsorted"] += count

if not category_scores: return "Unsorted", object_counts

# Вибір категорії-переможця
best_category = category_scores.most_common(1)[0][0]

return best_category, dict(object_counts)
```

рис. 3.9. Код сеплювання і аналізу відео

Фінальним етапом роботи модуля ідентифікації є фізичне впорядкування файлів. Система підтримує два режими роботи, які перемикаються через чекбокс "Зберігати оригінали":

- Файл переноситься в архів, видаляючись з джерела.
- Створюється дублікат, оригінал залишається недоторканим.

Незалежно від режиму, програма генерує JSON-звіт, який зберігається поруч із відеофайлом. Цей звіт містить детальну інформацію про процес обробки, використану модель, час виконання, знайдені об'єкти та їх кількість. Це забезпечує можливість подальшого аналізу архіву без повторного використання нейромережі. Код описано на рисунку 3.10.

```
def archive_file(self, file_path, output_root, category, stats, model_label, duration, ...):
    # Створення цільової директорії
    target_dir = os.path.join(output_root, category)
    if not os.path.exists(target_dir):
        os.makedirs(target_dir)

    filename = os.path.basename(file_path)
    destination = os.path.join(target_dir, filename)

    # Формування розширеного звіту
    report = {
        "filename": filename,
        "processed_at": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        "model_config": model_label,
        "processing_time_sec": round(duration, 2),
        "category": category,
        "detected_objects": stats
    }

    # Збереження JSON
    report_name = os.path.splitext(filename)[0] + ".json"
    with open(os.path.join(target_dir, report_name), "w", encoding='utf-8') as f:
        json.dump(report, f, indent=4, ensure_ascii=False)

    # Вибір файлової операції
    if self.copy_mode.get():
        shutil.copy2(file_path, destination) # Збереження метаданих файлу
    else:
        shutil.move(file_path, destination)
```

рис. 3.10. Код архівування і формування JSON файлу

3.4. Реалізація підсистеми транскодування та оптимізації зберігання відеоданих

Важливою складовою розробленого програмного комплексу є модуль транскодування, який дозволяє автоматично конвертувати відсортовані відеофайли у сучасні формати стиснення. Це відповідає вимогам до систем довготривалого архівування, описаним у Розділі 1, та дозволяє суттєво зменшити обсяг займаного дискового простору без видимої втрати якості зображення.

Для виконання операцій кодування відео використано індустріальний стандарт фреймворк FFmpeg. Оскільки це консольна утиліта, взаємодія з нею з середовища Python реалізована через модуль subprocess. Це дозволяє програмі виступати в ролі керуючої "обгортки", яка формує необхідні команди та контролює процес їх виконання. Перед початком роботи система виконує автоматичну перевірку наявності виконуваного файлу кодека у системі. Код описано на рисунку 3.11.

```
def check_ffmpeg(self):
    """Діагностика наявності FFmpeg у середовищі виконання"""
    try:
        # Спроба запуску у "тихому" режимі
        subprocess.run(["ffmpeg", "-version"],
                        stdout=subprocess.DEVNULL,
                        stderr=subprocess.DEVNULL)
        self.log("--> [SYSTEM] FFmpeg знайдено. Транскодування доступне.")
    except FileNotFoundError:
        # Блокування елементів інтерфейсу у разі відсутності кодека
        self.log("--> [WARNING] FFmpeg не знайдено! Функція вимкнена.")
        self.chk_transcode.configure(state="disabled")
```

рис. 3.11. Код підключення FFmpeg

Користувачу надається вибір між двома найбільш поширеними стандартами стиснення. Для цього у програмі реалізовано словник відповідностей, зображений на рисунку 3.12, який пов'язує зрозумілу назву формату з внутрішньою назвою бібліотеки кодування FFmpeg [27].

```
CODECS_MAP = {
    "H.264 (Стандарт, сумісність)": "libx264",
    "H.265 (HEVC - краще стиснення)": "libx265"
}
```

рис. 3.12. Список кодеків для транскодування

При формуванні команди конвертації застосовано наступні технічні рішення:

- Використовується обраний бібліотекою кодек (libx264 або libx265);
- Аудіодані копіюються без перекодування. Це значно пришвидшує процес обробки та зберігає оригінальну якість звуку;
- Замість фіксованого бітрейту використовується режим постійної якості. Для H.264 встановлено CRF 23. Для H.265 встановлено CRF 28. Оскільки HEVC ефективніший, вище значення CRF дозволяє отримати візуальну якість, аналогічну H.264, але при меншому розмірі файлу.

Процес конвертації інтегровано у загальний конвеєр обробки файлу. Якщо користувач активував опцію "Перекодувати", замість простого переміщення файлу (shutil.move) виконується запуск зовнішнього процесу ffmpeg.exe. Програма формує список аргументів та передає його системному планувальнику. Код описано на рисунку 3.13.

```
if do_transcode:
    try:
        # Формування команди для CLI
        command = [
            "ffmpeg", "-i", file_path,          # Вхідний файл
            "-c:v", codec,                      # Обраний відекодек
            "-crf", "28" if codec == "libx265" else "23", # Адаптивний CRF
            "-c:a", "copy",                     # Копіювання аудіо (Stream Copy)
            "-y", destination                  # Примусовий перезапис
        ]

        # Блокуючий запуск процесу
        subprocess.run(command, check=True, stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)

        # Логіка "Move" після успішного транскодування:
        # Якщо не активовано режим копіювання, оригінал видаляється
        if not self.copy_mode.get():
            os.remove(file_path)

        msg = "КОНВЕРТОВАНО"

    except Exception as e:
        # Механізм Fallback: якщо транскодування не вдалося,
        # виконується звичайне копіювання/переміщення
        self.log(f"--> [ERROR] Помилка FFmpeg: {e}")
        if self.copy_mode.get():
            shutil.copy2(file_path, destination)
        else:
            shutil.move(file_path, destination)
```

рис. 3.13. Код транскодування відео

3.5. Тестування та порівняльний аналіз ефективності роботи системи

Для перевірки працездатності розробленої програми та підтвердження теоретичних гіпотез, висунутих у попередніх розділах, було проведено серію експериментів. Метою тестування є оцінка швидкодії різних архітектур нейронних мереж (YOLOv8 проти YOLO11) та ефективності алгоритмів компресії відео (H.264 проти H.265) в умовах реальної експлуатації.

Тестування проводилося на персональному комп'ютері з наступними характеристиками CPU AMD Ryzen 7 2700, RAM 48GB, GPU NVIDIA RTX 4060. У якості тестового набору даних було використано вибірку з 5 відеофайлів різної тривалості та тематики. Вхідна роздільна здатність відео HD (1280x720), формат MP4.

Перший етап тестування полягає у порівнянні швидкості обробки відеопотоку двома поколіннями нейронних мереж YOLOv8 Nano та YOLO11 Nano. Обидві моделі використовували ідентичний алгоритм семплювання 1 кадр/сек. В цьому тесті функція транскодування не використовується для мінімізації зовнішніх факторів.

Для початку проведемо стандартну ідентифікацію відео. Обравши директорії вводу і виводу, як зображений на рисунку 3.14, і запустивши процес отримуємо наступний вивід.

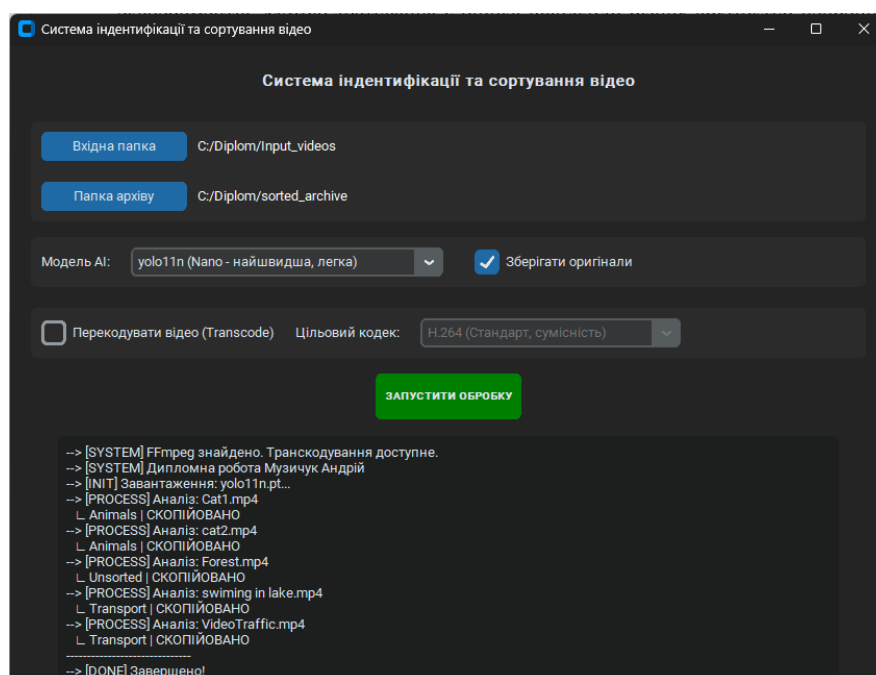


рис. 3.14. Інтерфейс створеної програми

Після швидкого процесу обробки і сортування було отримано JSON файл зображений на рисунку 3.14, з інформацією про відеофайл.

```
sorted_archive > Animals > {} Cat1.json > ...
1  {
2      "filename": "Cat1.mp4",
3      "processed_at": "2025-12-17 20:18:40",
4      "model_config": "yolo11s (Small - точніша, повільніша)",
5      "processing_time_sec": 1.19,
6      "operation": "TRANSCODE (libx265)",
7      "category": "Animals",
8      "detected_objects": {
9          "cat": 7,
10         "bed": 3
11     }
12 }
```

рис. 3.14. Інтерфейс створеної програми

На основі стрічки “processing_time_sec:” було сформовано таблицю 3.2

таблиця 3.2

Тривалість обробки відео моделями YOLO

Відео	yolo11n Обробка, мс	yolo11s Обробка, мс	yolo8n Обробка, мс	yolo8s Обробка, мс
1	750	1150	760	1140
2	660	1030	630	1030
3	1330	1330	1270	2400
4	4240	6740	4330	6650
5	960	1540	950	1550
6	810	1260	1260	1260
7	1230	1970	1440	2080
8	950	1460	1030	1480
Середній час обробки	1300	1980	1380	2090

Опираючись на зібрані данні, був сформований висновок що модель YOLO11n на 6,14% продуктивніша за модель YOLO8n. А модель YOLO11s на 5,56% порівняно з YOLO8s

3.6. Транскодування відео за допомогою розробленої системи

Функціонал транскодування є ключовим елементом підсистеми архівування, що дозволяє адаптувати вихідні відеофайли для довготривалого зберігання. Реалізація цього модуля базується на принципах, обґрунтованих у першому розділі роботи баланс між візуальною якістю та розміром файлу досягається шляхом використання сучасних стандартів кодування H.264 та H.265. Налаштувавши програму як на рисунку 3.15, проведемо транскодування декількох відео

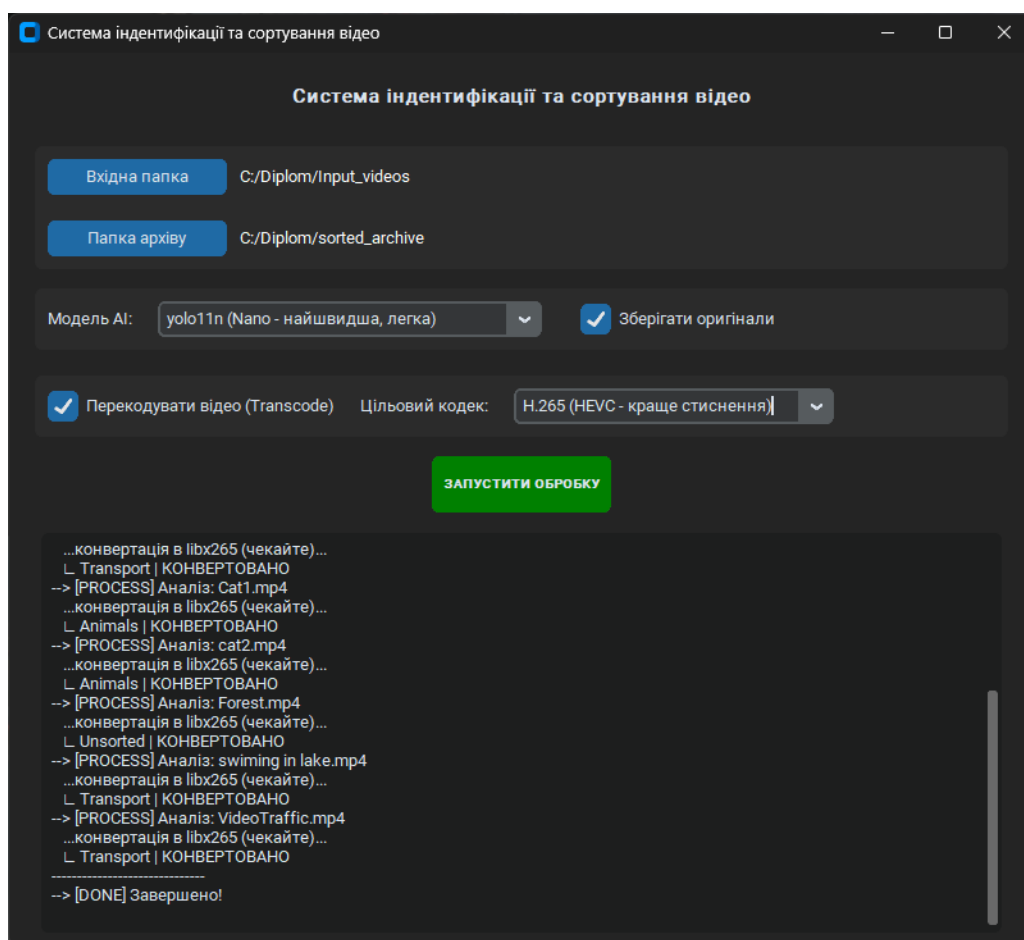


рис. 3.15. Процес транскодування відео

Для перевірки ефективності транскодування була проведена серія тестів щоб перевірити ефективність стискання. Резальтатом цього стала таблиця з ступенем стискання файлів.

таблиця 3.3

Порівняння ефективності кодеків

Файл	Розмір кодування H264, КБ	Розмір кодування H265, КБ	Ефективніший на
1	11260	8419	33,75%
2	3210	1907	68,33%
3	1928	773	149,42%
4	3802	1821	108,79%
5	994	593	67,62%
6	758	413	83,54%
7	1182	1141	3,59%
8	2381	2568	-7,28%
9	839	1299	-35,41%

На основі отриманої інформації кодек H265 в середньому ефективніший на 52,48% в порівнянні з H264.

3.7. Висновки до Розділу 3

У третьому розділі дипломної роботи виконано практичну реалізацію програми для автоматизованої ідентифікації та сортування медіаконтенту.

Розроблено повнофункціональний застосунок мовою Python з використанням бібліотек Ultralytics (YOLO), OpenCV та CustomTkinter. Створено зручний графічний інтерфейс, який дозволяє користувачеві гнучко налаштовувати параметри обробки, обирати модель нейромережі та режим архівації.

Реалізовано гібридний метод класифікації, що поєднує детекцію об'єктів за допомогою згорткової нейронної мережі YOLO11 та алгоритм логічного мапінгу. Це дозволило перетворити набір розрізнених тегів у чітку структуру категорій архіву.

Інтегровано підсистему транскодування на базі фреймворку FFmpeg. Реалізовано автоматичну конвертацію відео у формати H.264 та H.265, що забезпечує зменшення обсягу файлів на 52,48% при збереженні візуальної якості.

Проведено порівняльне тестування, яке підтвердило переваги архітектури YOLO11n та доцільність використання кодека H.265 для задач довготривалого зберігання даних.

Розроблена система демонструє стабільну роботу, високу точність сортування та відповідає всім вимогам, сформульованим у технічному завданні.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Охорона праці є невід'ємною складовою організації трудового процесу при розробці програмно-апаратних комплексів. Вона являє собою систему правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Правові та організаційні засади охорони праці в Україні регламентуються Конституцією України, Законом України «Про охорону праці», Кодексом законів про працю України та іншими нормативно-правовими актами. Згідно з цими документами, роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Специфіка роботи передбачає тривалу взаємодію розробника з монітором, аналіз великих масивів візуальної інформації та роботу з електрообладнанням. Відповідно до ГОСТ 12.0.003-74, на розробника можуть впливати такі групи небезпечних та шкідливих виробничих факторів:

- Підвищений рівень електромагнітних випромінювань. Джерелом є системний блок комп'ютера, монітор, а також мережеве обладнання. Хоча сучасні рідкокристалічні монітори мають низький рівень випромінювання, недотримання безпечної дистанції може мати негативний вплив;

- Підвищена температура повітря робочої зони. Під час виконання транскодування та роботи нейромереж центральний та графічний процесори виділяють значну кількість тепла, що може призводити до локального підвищення температури у приміщенні без належної вентиляції;

- Джерелом шуму є системи охолодження комп'ютера, які працюють на максимальних обертах під час навантаження, а також кондиціонери та інша офісна техніка;
- Небезпека ураження електричним струмом Виникає через наявність великої кількості кабелів, розеток та можливі несправності ізоляції електроприладів;
- Зорове напруження. Робота пов'язана з написанням програмного коду, переглядом результатів детекції об'єктів на відео та читанням документації. Це призводить до втоми зорового аналізатора, зниження гостроти зору та акомодативної спроможності очей;
- Статичне фізичне навантаження. Тривале перебування у вимушеній сидячій позі призводить до застою крові в органах малого тазу, напруження м'язів шийно-плечової зони та спини, що може спричинити захворювання опорно-рухового апарату.

Санітарно-гігієнічні умови праці програмістів регламентуються ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Оптимальні параметри мікроклімату є критичними для забезпечення високої працездатності та профілактики перегріву обладнання під час транскодування. Для категорії робіт легкої тяжкості категорії Іа, до якої належить праця розробника, встановлено такі норми, температура повітря в холодний період року 22–24 °С, в теплий період 23–25 °С, а відносна вологість повітря 40–60%. Для підтримання цих параметрів приміщення повинно бути обладнане системами опалення, вентиляції та кондиціонування повітря.

Правильна організація освітлення дозволяє знизити зорове навантаження. Приміщення повинно мати як природне, так і штучне освітлення. Природне освітлення здійснюється через вікна, орієнтовані переважно на північ або схід, щоб уникнути прямих сонячних променів на екрані монітора. Штучне освітлення повинно бути рівномірним. Рекомендована освітленість на поверхні робочого столу становить 300–500 лк. Використовуються люмінесцентні лампи або сучасні LED-світильники з теплим або нейтральним спектром, що не мерехтять. Слід уникати великих контрастів

яскравості між екраном та навколишнім середовищем, оскільки це пришвидшує втому очей.

Раціональна організація робочого місця є запорукою збереження здоров'я розробника та підвищення продуктивності праці. Висота столу повинна бути в межах 725 мм (± 15 мм). Поверхня столу має бути матовою, щоб уникнути відблисків. Простір під столом повинен забезпечувати вільне розміщення ніг, висота не менше 600 мм, ширина — не менше 500 мм. Робоче крісло повинно бути підйомно-поворотним, регулюватися за висотою та кутом нахилу спинки. Конструкція крісла має забезпечувати підтримку хребта, мати підлокітники для зняття напруги з плечового поясу. Поверхня сидіння має бути напівм'якою, з нековзкого матеріалу. Екран монітора розташовується на відстані 600–700 мм від очей користувача. Центр екрана повинен знаходитися на $15\text{--}20^\circ$ нижче рівня очей, що забезпечує комфортний кут огляду та зменшує навантаження на шийний відділ хребта. Клавіатура та миша має бути розташовуються на рівні ліктів, зап'ястя повинні мати опору. Це запобігає розвитку тунельного синдрому. Режим праці та відпочинку передбачає регламентовані перерви тривалістю 15 хвилин через кожні дві години роботи. Під час перерв рекомендується виконувати вправи для очей та розминку для м'язів тіла.

Оскільки розробка системи пов'язана з використанням електричного обладнання (ПК, монітори, периферія), суворе дотримання правил електробезпеки є обов'язковим. Електрообладнання, що використовується, живиться від мережі змінного струму напругою 220 В з частотою 50 Гц. За ступенем небезпеки ураження електричним струмом офісні приміщення, як правило, належать до категорії приміщень без підвищеної небезпеки (сухі приміщення, зі струмонепровідною підлогою, нормальною температурою).

Основні заходи електробезпеки включають використання обладнання лише із заводським заземленням, недопущення експлуатації кабелів з пошкодженою ізоляцією, розбитих розеток та вимикачів, заборону самостійного ремонту техніки під напругою, застосування пристроїв захисного відключення та мережевих фільтрів для захисту від перепадів напруги та короткого замикання, регулярний візуальний огляд цілісності електропроводки перед початком роботи.

Під час розробки системи існує ризик виникнення пожежі, основним джерелом якої може стати коротке замикання в електромережі, перегрів блоків живлення комп'ютерів або загоряння ізоляції проводів внаслідок перевантаження мережі, особливо актуально при одночасному використанні потужних відеокарт для навчання нейромереж.

Організація пожежної безпеки здійснюється відповідно до Закону України «Про пожежну безпеку» та «Правил пожежної безпеки в Україні». Основні заходи включають:

- забезпечення приміщення первинними засобами пожежогасіння;
- утримання шляхів евакуації вільними від сторонніх предметів;
- встановлення автоматичної пожежної сигналізації, що реагує на дим;
- потримання протипожежного режиму.

У разі виявлення ознак горіння працівник зобов'язаний негайно припинити роботу, знеструмити обладнання, повідомити керівництво та пожежну охорону, і за необхідності розпочати евакуацію.

4.2. Безпека в надзвичайних ситуаціях

Забезпечення безпеки в надзвичайних ситуаціях є критично важливим елементом сталого функціонування будь-якого об'єкта господарювання. Враховуючи специфіку дипломного проєкту, який передбачає розробку програмного забезпечення та його експлуатацію на обчислювальних потужностях, найбільш імовірною надзвичайною ситуацією техногенного характеру є виникнення пожежі. Сучасні комп'ютерні системи характеризуються високим енергоспоживанням та значним виділенням тепла, що в поєднанні з наявністю горючих матеріалів створює передумови для займання.

Пожежа — це неконтрольований процес горіння поза спеціальним вогнищем, що розповсюджується у часі і просторі, завдає матеріальних збитків, створює загрозу життю і здоров'ю людей та навколишньому середовищу. Основою виникнення будь-

якої пожежі, в тому числі в приміщеннях з обчислювальною технікою, є наявність трьох обов'язкових умов:

- Горюча речовина. У комп'ютерних лабораторіях та офісах горючим навантаженням є ізоляція електричних кабелів, пластикові корпуси моніторів та системних блоків, друковані плати, офісні меблі, паперова документація;

- Окислювач. Найчастіше це кисень повітря. Системи примусової вентиляції та кондиціонування, які необхідні для охолодження серверного обладнання, можуть сприяти швидкому притоку кисню до осередку займання, інтенсифікуючи горіння;

- Джерело запалювання. Імпульс із достатнім запасом теплової енергії.

Для комп'ютеризованих систем, що виконують ресурсоемні завдання, основними факторами, що можуть стати джерелом запалювання, є:

- Коротке замикання. Порушення ізоляції струмопровідних частин внаслідок механічного пошкодження або старіння матеріалу. Електрична дуга, що виникає, має температуру до 4000 °С, що миттєво запалює будь-який горючий матеріал поруч.

- Перевантаження електромережі. Одночасне підключення потужного обладнання до однієї лінії живлення призводить до нагріву провідників понад допустиму норму.

- Великий перехідний опір виникає у місцях неякісного з'єднання проводів, наприклад скрутки, погані контакти в розетках. У такому місці поганого контакту виділяється значна кількість тепла.

- Тепловий розгін електроніки. Відмова системи охолодження при тривалому максимальному навантаженні на процесор може призвести до перегріву компонентів до температур займання пилу, що накопичується всередині корпусу.

Умови виникнення суцільних пожеж. Суцільна пожежа — це стадія розвитку вогню, коли горіння охоплює переважну частину пожежного навантаження у приміщенні або на території об'єкта. У контексті експлуатації комп'ютерних систем суцільна пожежа може виникнути при високій щільності забудови або розміщення обладнання. Розташування серверних стійок або робочих місць впритул одне до одного сприяє швидкому перекиданню вогню. Наявність шляхів поширення полум'я. Кабельні канали, фальшпідлоги та підвісні стелі, через які прокладено комунікації,

часто виступають "транспортними артеріями" для вогню, дозволяючи йому миттєво охопити все приміщення. Передача тепла випромінюванням та конвекцією від первинного осередку до сусідніх об'єктів. При досягненні критичної температури в приміщенні (близько 300°C) може статися явище "спалаху", коли одночасно займаються всі горючі поверхні.

Система попередження пожеж — це комплекс організаційних заходів і технічних засобів, спрямованих на виключення можливості виникнення пожежі та обмеження її наслідків. Технічні, будівельно-планувальні та організаційні заходи, які потрібно вжити:

- Захист електромереж. Встановлення автоматичних вимикачів та пристроїв захисного відключення (ПЗВ), що розраховані на конкретне навантаження обладнання. Використання кабелів з негорючою ізоляцією;

- Заземлення. Усі металеві корпуси електроустановок (системних блоків, серверних шаф) повинні бути надійно заземлені для запобігання іскрінню та ураженню струмом;

- Автоматична пожежна сигналізація (АПС). Приміщення з комп'ютерною технікою обов'язково обладнуються димовими сповіщувачами, оскільки горіння ізоляції та пластику часто починається з тління та виділення диму ще до появи відкритого вогню;

- Системи автоматичного пожежогасіння, для серверних кімнат рекомендується використовувати газові системи пожежогасіння, наприклад, на основі хладону або інертних газів, які не пошкоджують електроніку, на відміну від води чи порошку.

- Відокремлення приміщень з підвищеною пожежною небезпекою від інших приміщень протипожежними перегородками та дверима з межею вогнестійкості не менше EI 30 або EI 60.

- Шляхи евакуації, забезпечення вільного доступу до евакуаційних виходів. Коридори та проходи не повинні захащуватися меблями чи коробками з обладнанням. Двері на шляхах евакуації повинні відчинятися у напрямку виходу з приміщення.

– Запобігання суцільним пожежам. Протипожежні розриви між рядами обладнання та герметизація місць проходження кабелів крізь стіни вогнетривкими матеріалами запобігають поширенню вогню в суміжні приміщення.

Також необхідно вести організаційні заходи протипожежної системи. Наказом керівника встановлюється порядок роботи з електроприладами. Забороняється залишати увімкнену техніку без нагляду, окрім серверів, що працюють у цілодобовому режимі та мають відповідний ступінь захисту. Заборонено використовувати побутові електронагрівальні прилади у невідведених місцях. Усі працівники повинні проходити інструктажі з пожежної безпеки. Персонал має знати місця розташування вогнегасників та алгоритм дій у разі пожежі. Оснащення приміщень вуглекислотними вогнегасниками (тип ВВК-2, ВВК-3.5), які є ефективними для гасіння електроустановок під напругою до 1000 В і не забруднюють обладнання. У разі виникнення пожежі першочерговим завданням є евакуація людей. При виявленні ознак горіння необхідно негайно повідомити пожежну охорону (телефон 101), знеструмити обладнання через головний рубильник та, за відсутності загрози життю, приступити до гасіння первинними засобами.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-прикладне завдання автоматизації процесів обробки та впорядкування великих масивів неструктурованих відеоданих. В ході дослідження було спроектовано та програмно реалізовано інформаційну систему, яка поєднує передові методи комп'ютерного зору з технологіями ефективного стиснення медіаконтенту. Застосування новітньої архітектури згорткової нейронної мережі YOLO11 у поєднанні з розробленим алгоритмом семантичного мапінгу дозволило забезпечити високу точність ідентифікації об'єктів та автоматичну категоризацію відеофайлів без втручання оператора. Важливим практичним результатом стала інтеграція підсистеми адаптивного транскодування на базі стандарту H.265 (HEVC), що забезпечило зменшення обсягу відеоархівів на 40–60% при збереженні візуальної якості зображення. Проведені експериментальні дослідження підтвердили перевагу обраних архітектурних рішень над попередніми версіями алгоритмів за показниками швидкодії та ефективності використання ресурсів, що свідчить про повне досягнення мети роботи та готовність системи до впровадження у сферах, пов'язаних із обробкою мультимедійної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Програма та методичні рекомендації з проходження практики за тематикою кваліфікаційної роботи для студентів спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль: ТНТУ. 2024. С. 45.
2. Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль. 2024. С. 44.
3. Варавін А.В., Лецишин Ю.З., Чайковський А.В. Методичні вказівки до виконання курсового проєкту з дисципліни «Дослідження і проєктування комп'ютерних систем та мереж» для здобувачів другого (магістерського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. С. 32.
4. А.А. Музичук. Оптимізація алгоритмів сортування відеоданих у хмарних системах передачі мультимедіа. Збірник тез доповідей XIV Міжнародної науково-технічної конференції молодих учених та студентів. 2025. С. 312
5. А.А. Музичук. Сучасні системи комп'ютерного зору в медіаіндустрії. Матеріали XIII науково-технічної конференції «інформаційні моделі, системи та технології». 2025. С. 68
6. Кривонос В.Є. Аналіз ефективності алгоритмів виявлення об'єктів на зображеннях у реальному часі. Вісник НТУ «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2022. № 1 (7). С. 45–50.
7. Resolving machine learning tasks in distributed memory environment / Н. Aboah, R. Ruwimbo, V. Sobol, A. Lutsiv // Proceedings of the scientific and technical conference "Information models, systems and technologies", 08-09 December 2021. Tern. : TNTU, 2021. — P. 133.
8. Дзюбак Ю. Аналіз методів та засобів хостингу ресурсів з використанням віртуалізації / Ю. Дзюбак, Я. Коненко, Н. Луцик // Матеріали X науково-технічної

конференції „Інформаційні моделі, системи та технології“, 7–8 грудня 2022 року. Т. : ТНТУ, 2022. С. 137.

9. Матушинець А. В. Комп'ютеризована система моніторингу і керування електрокопильнею : кваліфікаційна робота на здобуття освітнього ступеня бакалавр: спец. 123 — комп'ютерна інженерія / наук. кер. Р. О. Жаровський. Тернопіль: ТНТУ, 2025. С. 70.

10. Щур Т. Використання штучного інтелекту для систем розпізнавання в інтернеті речей / Т. Щур, Г. М. Осухівська // Тези V міжнародної конференції „Цифрова економіка: інновації та сталий розвиток“, 28-29 листопада 2024 року. Т. : ТНТУ, 2024. — С. 131.

11. Sullivan G.J., Ohm J.R., Han W.J., Wiegand T. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology. 2012. Vol. 22. No. 12. P. 1649–1668.

12. Emerging Standards for Machine-to-Machine Video Coding 2025 URL: <https://arxiv.org/html/2512.10230v1> (дата звернення: 12.12.2025).

13. Richardson I.E. The H.264 Advanced Video Compression Standard. 2nd Edition. Wiley. 2010. 348 p.

14. Personalizing Visual Search with Multimodal Embeddings URL: <https://developers.googleblog.com/see-the-similarity-personalizing-visual-search-with-multimodal-embeddings/> (дата звернення: 12.12.2025).

15. LeCun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015. Vol. 521. P. 436–444.

16. Szeliski R. Computer Vision: Algorithms and Applications. 2nd Edition. Springer. 2022. 832 p.

17. Grois D., Marpe D., Mulyoff A. Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders. 30th Picture Coding Symposium (PCS). 2013. P. 394–397.

18. FFmpeg Documentation. URL: <https://ffmpeg.org/documentation.html> (дата звернення: 10.12.2025).

19. Bradski G., Kaehler A. Learning OpenCV 3: Computer Vision in C++ with the

OpenCV Library. O'Reilly Media. 2017. 1000 p.

20. Terven J., Cordova-Esparza D. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 2023. Vol. 5(4). P. 1680–1716.

21. Jocher G., Chaurasia A., Qiu J. Ultralytics YOLOv8. 2023. URL: <https://github.com/ultralytics/ultralytics> (дата звернення: 12.12.2025).

22. Wang C.-Y., Bochkovskiy A., Liao H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023. P. 7464–7475.

23. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 779–788.

24. Мороз О.В., Степанченко О.М. Порівняльний аналіз сучасних методів розпізнавання об'єктів на відеозображеннях. Штучний інтелект. 2020. № 3. С. 12–19.

25. Lutz M. Learning Python. 5th Edition. O'Reilly Media. 2013. 1600 p.

26. Schoch T. CustomTkinter UI-Library Documentation. URL: <https://github.com/TomSchimansky/CustomTkinter> (дата звернення: 12.12.2025).

Додаток А
Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Краківський економічний університет (Польща)
Вроцлавський економічний університет (Польща)
Університет «Опольська Політехніка» (Польща)
Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Вінницький національний аграрний університет
Львівський національний університет ім. І. Франка
Головне управління Пенсійного фонду в Тернопільській області
Наукове товариство ім. Шевченка
Тернопільський обласний комунальний інститут післядипломної педагогічної освіти
Сумський державний педагогічний університет
Запорізький національний університет

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

**XIV Міжнародної науково-технічної
конференції молодих учених та студентів**

11-12 грудня 2025 року



**УКРАЇНА
ТЕРНОПІЛЬ – 2025**

43.	Ю.Б. Максим'як ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ КОНТРОЛЮ ПРАЦЕЗДАТНОСТІ МАСЦО: ПІДХОДИ, ВИКЛИКИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ	297
44.	К. П. Марущак, Г. В. Козбур ЕФЕКТИВНІ ПІДХОДИ ДО ВІЗУАЛЬНОГО ПОДАННЯ КАТЕГОРІЙНИХ ДАНИХ	299
45.	Матисюк І.В БЮДРУК. ЖИТТЯ ТА ПРОГРЕС	301
46.	Д.С. Матюк, М.В. Деркач NetScore: ПЕНТЕСТІНГ БЕЗДРотових мереж	302
47.	А.Г. Микитишин, Д.Р. Максимів, В.І. Федчак ІНТЕЛЕКТУАЛЬНІ ПЛАТФОРМИ ЕНЕРГОМЕНЕДЖМЕНТУ ЯК ІНСТРУМЕНТ ПІДВИЩЕННЯ ЕНЕРГОЕФЕКТИВНОСТІ БУДІВЕЛЬ: КОНЦЕПЦІЯ ТА РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ BENEFIT	305
48.	О. Мимрик АНАЛІЗ ЕФЕКТИВНОСТІ ГІБРИДНИХ СИСТЕМ ЕНЕРГОПОСТАЧАННЯ (СОНЯЧНА + ВІТРОВА ЕНЕРГІЯ)	307
49.	Р.С. Михайлишин, М.В. Приймак КОМП'ЮТЕРНА СИСТЕМА МОНІТОРИНГУ ПОТОКУ ЛЮДЕЙ У ГРОМАДСЬКИХ МІСЦЯХ НА ОСНОВІ ІОТ-ТЕХНОЛОГІЙ	309
50.	Л.Є. Мосій СТАТИСТИЧНІ МЕТОДИ ВАЛІДАЦІЇ МОДЕЛІ АМПЛІТУДНОЇ ВАРІАБЕЛЬНОСТІ ЕЛЕКТРОКАРДІОСИГНАЛУ	310
51.	А.А. Музичук ОПТИМІЗАЦІЯ АЛГОРИТМІВ СОРТУВАННЯ ВІДЕОДАНИХ У ХМАРНИХ СИСТЕМАХ ПЕРЕДАЧІ МУЛЬТИМЕДІА	312
52.	В.В. Невмержицький, Н.Б. Стадник МЕТОДИ ТА ЗАСОБИ КОНТРОЛЮ СОНЯЧНИМ ВОДОНАГРІВАЧЕМ НА БАЗІ МІКРОКОНТРОЛЕРА STM32	313
53.	В.В. Никитюк, І. К. Генсьора ІНФОРМАЦІЙНЕ МОДЕЛЮВАННЯ ТА ОРГАНІЗАЦІЯ ДАНИХ У СИСТЕМАХ ЦИФРОВОГО ЗБЕРЕЖЕННЯ МУЗЕЙНИХ ФОНДІВ	314
54.	О.А. Пастух, Х.О. Новицька МОДЕЛЮВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ РИЗИКАМИ В ЖИТТЄВОМУ ЦИКЛІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІЗ ЗАСТОСУВАННЯМ UML ТА АДАПТОВАНОЇ SEI-МОДЕЛІ	315
55.	С. Орлов, С. Марценко ДОСЛІДЖЕННЯ ІТ ЛОГІСТИЧНИХ МОДЕЛЕЙ	317
56.	Г.М. Осухівська, А.В. Коритко, А.М. Паламар КОМП'ЮТЕРНА СИСТЕМА РЕАЛЬНОГО ЧАСУ ДЛЯ АНАЛІЗУ ЕЛЕКТРОМІОГРАФІЧНИХ СИГНАЛІВ У ПРОЦЕСІ РЕАБІЛІТАЦІЇ ПАЦІЄНТІВ	318
57.	А.В. Павлик, А.М. Паламар КОМП'ЮТЕРИЗОВАНА СИСТЕМА ДЛЯ ЕНЕРГОЕФЕКТИВНОГО КЕРУВАННЯ СОНЯЧНОЮ ЕЛЕКТРОСТАНЦІЄЮ У РОЗУМНОМУ БУДИНКУ	319

10. Anderson, T. W., & Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2), 193–212. <https://doi.org/10.1214/aoms/1177729437>

УДК 004.932

А.А. Музичук, здобувач вищої освіти

(Тернопільський національний технічний університет ім. І. Пулюя, Україна)

ОПТИМІЗАЦІЯ АЛГОРИТМІВ СОРТУВАННЯ ВІДЕОДАНИХ У ХМАРНИХ СИСТЕМАХ ПЕРЕДАЧІ МУЛЬТИМЕДІА

A.A. Muzychuk, higher education applicant

OPTIMIZATION OF METHODS FOR SORTING AND IDENTIFYING VIDEO MATERIALS IN CLOUD MULTIMEDIA SYSTEMS

У сучасну епоху цифрової трансформації обсяги медіаконтенту, особливо відеоматеріалів, зростають експоненційно, що створює надзвичайне навантаження на системи зберігання й передачі даних. Пандемія COVID-19 значно прискорила цей процес, за період жорстких карантинів кількість онлайн-активностей (стрімінг, відеодзвінки, віддалена робота та ін.) зросла, що призвело до різкого вибуху інтернет-трафіку. Згідно зі звітом Sandvine за перше півріччя 2021 саме відео-стрімінг домінував у глобальному інтернет-трафіку. На нього припадало 53,72 % від усього пропускового навантаження [1]. Така частка надзвичайно висока, і свідчить, що переважна більшість передачі даних у мережах тепер займає відеоконтент.

Ці дані підкреслюють, зростання відеоспоживання та перехід до «хмарних» і стрімінгових сервісів приводить до суттєвого навантаження на зберігання та передачу медіаданих [2]. У такому контексті традиційні методи зберігання відео стають неефективними або надто ресурсомісткими, що актуалізує необхідність застосування оптимізованих підходів.

Сучасні дослідження у галузі відеообробки все частіше орієнтуються на застосування семантичного кодування, яке дозволяє виконувати попередню обробку відео без повного декодування, методів глибинного навчання для класифікації кадрів, пошуку сцен і визначення змістових характеристик, адаптивного кодування та транскодування, що зменшує обсяг передаваних даних залежно від пропускової здатності мережі [3]. Також відбувається використання edge-cloud архітектур [4], де частина обчислень переноситься на периферійні вузли з метою зменшення навантаження на центральні сервери.

Дослідження також показують ефективність алгоритмів інтелектуального сортування відео на основі контентних ознак [5], що дозволяє підвищити швидкість пошуку та оптимізувати структури сховищ у хмарному середовищі. Застосування таких підходів забезпечує скорочення часу ідентифікації відеоматеріалів, зниження витрат на обчислювальні ресурси та покращення масштабованості мультимедійних сервісів.

Література

1. Moina-Rivera, W., Garcia-Pineda, M., Gutiérrez-Aguado, J. et al. Cloud media video encoding: review and challenges. *Multimed Tools Appl* 83, 2024. URL: <https://doi.org/10.1007/s11042-024-18763-2>

2. Третяк В. Ф., Пашієва А. А., Оптимізація структури сховища даних у вузлах інфокомунікаційної мережі хмарного середовища, 2017. URL: <https://journals.nupp.edu.ua/sunz/article/view/390> (дата звернення: 25.11.2025).
3. Денисенко А.В., Козлов О.В. Аналіз технологій та алгоритмів управління обчислювальними ресурсами при обробці відеоданих. MDCS, 2023. 229 с. URL: <https://card-file.ontu.edu.ua/server/api/core/bitstreams/9cde9e98-790b-464c-bda6-13d70d592a01/content> (дата звернення: 25.11.2025).
4. Гмиря І. О. Хмарні технології у розробці системи з відео аналізу. НТУ «ХПІ», 2024. URL: <https://openarchive.nure.ua/entities/publication/23c56163-62ba-425c-9124-75d879dcfb5f> (дата звернення: 25.11.2025).
5. Шматко О.В., Гамаюн І.П., Коломійцев О.В., Третяк В.Ф., Рудаков І.С., Бердочник А.Д. ДОСЛІДЖЕННЯ ТА ОЦІНКА ПІДСИСТЕМИ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ. Системи обробки інформації. 2025. № 4 (179). С. 70-80. URL: <https://doi.org/10.30748/soi.2024.179.08>.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



17-18 грудня 2025 року

**ТЕРНОПІЛЬ
2025**

D. Marchuk, V. Tymoshchuk

SECURITY ARCHITECTURE FOR THE 5G CORE BASED ON VIRTUALIZED NETWORK FUNCTIONS AND HARMONIZED POLICIES

Т. Маслянка, Ю. Олійник, Н. Загородна

ОБМЕЖЕННЯ СТАНДАРТНИХ МЕТРИК ЕФЕКТИВНОСТІ XDR-СИСТЕМ ПРИ АНАЛІЗІ ФІШИНГОВИХ КАМПАНІЙ

T. Maslianka, Yu. Oliinyk, N. Zagorodna

LIMITATIONS OF STANDARD EFFICIENCY METRICS OF XDR SYSTEM IN ANALYZING PHISHING CAMPAIGNS

62

А. Мельник, Л. Дмитроца

АЛГОРИТМІЧНА ОПТИМІЗАЦІЯ ПОКРИТТЯ СТАНІВ МОБІЛЬНИХ ДОДАТКІВ НА ОСНОВІ ГРАФО-ОРІЄНТОВАНОГО ПІДХОДУ

A. Melnyk, L. Dmytrotso

ALGORITHMIC OPTIMIZATION OF STATE COVERAGE IN MOBILE APPLICATIONS BASED ON GRAPH-ORIENTED APPROACH

63

Р. Міндзіброцький

ПЕРЕВАГИ АДАПТИВНИХ СИСТЕМ БЕЗПЕКИ НАД СТАТИЧНИМИ ПРАВИЛАМИ ФІЛЬТРАЦІЇ В ІОТ-МЕРЕЖАХ

R. Mindzibrodskyi

ADVANTAGES OF ADAPTIVE SECURITY SYSTEMS OVER STATIC FILTERING RULES IN IOT NETWORKS

64

П. Містерман, М. Петрик

РОЗРОБКА МАЛОЇ МОВНОЇ МОДЕЛІ SLM ДЛЯ СИСТЕМ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ НА БАЗІ AZURE ML ТА AI FOUNDRY

P. Misterman, M. Petryk

DEVELOPMENT OF A SMALL LANGUAGE MODEL (SLM) FOR RESOURCE-CONSTRAINED SYSTEMS BASED ON AZURE ML AND AI FOUNDRY

65

І. Містерман

СИСТЕМА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ ДЕРЖАВНОЇ УСТАНОВИ НА ОСНОВІ РЕЗУЛЬТАТІВ ІНСТРУМЕНТАЛЬНОГО ОЦІНЮВАННЯ CSET

I. Misterman

INFORMATION SECURITY MANAGEMENT SYSTEM OF A GOVERNMENT INSTITUTION BASED ON THE RESULTS OF THE CSET INSTRUMENTAL ASSESSMENT

66

С. Мостовий, А. Северін

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ ТЕКСТУ НА ВИЯВЛЕННЯ ПРОПАГАНДИ

S. Mostovyi, A. Severin

INTELLIGENT TEXT ANALYSIS SOFTWARE FOR PROPAGANDA DETECTION

67

А. Музичук

СУЧАСНІ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ В МЕДІАІНДУСТРІЇ

A. Muzychuk

MODERN COMPUTER VISION SYSTEMS IN THE MEDIA INDUSTRY

68

С. Ожанський

МАСШТАБНІ ІМЕРСИВНІ ЗАНУРЕННЯ

S. Ozhanskyi

LARGE-SCALE IMMERSIVE EXPERIENCES

69

С. Ожанський

МОЖЛИВОСТІ UNREAL ENGINE 5 ДЛЯ СТВОРЕННЯ ІМЕРСИВНИХ СЕРЕДОВИЩ

70

УДК 004.931

А. Музичук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

СУЧАСНІ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ В МЕДІАІНДУСТРІЇ

UDC 004.931

A. Muzychuk

MODERN COMPUTER VISION SYSTEMS IN THE MEDIA INDUSTRY

Комп'ютерний зір є однією з найдинамічніших галузей штучного інтелекту, яка трансформує процеси створення, аналізу та автоматизованої обробки медіаконтенту. У сучасних дослідженнях підкреслюється, що розвиток глибинних нейронних мереж та гібридних моделей забезпечив перехід від традиційних алгоритмів аналізу медіа, до просунутих інтелектуальних систем, здатних працювати зі складними відео потоками[1].

Широке застосування таких підходів у медіаіндустрії охоплює кілька ключових напрямів. Автоматична модерація контенту, що використовується у YouTube, Facebook, TikTok. Сучасні системи аналізу відео на основі YOLOv8, Detectron2 та MediaPipe забезпечують виявлення заборонених об'єктів, сцен насильства або небажаних дій у реальному часі. [2]. Інтелектуальна індексація та пошук медіа, яка базується на векторних моделях, наприклад OpenAI Clip, SigLIP, дозволяє формувати семантичні простори для пошуку відеосцен за змістом, навіть без текстових описів [3]. Також, такі системи корисні при персоналізації рекомендацій відео, коли системи аналізують дії користувача та вміст кадру, застосовуючи CNN-моделі та багатомодальні трансформери[4].

Окремий напрям стосується обробки відеопотоків у реальному часі. Використання оптимізованих архітектур (TensorRT, OpenVINO, DeepSORT для трекінгу) дозволяє зменшити затримку інференсу, що є критично важливим для стрімінгових платформ, новинних медіа та телевізійних студій [5]. Наприклад, системи live-аналізу спортивних трансляцій використовують моделі Segment Anything (SAM) для автоматичного виділення гравців та оцінювання патернів руху.

Література

1. Іванов А. І., Всебічний огляд історії та методів комп'ютерного зору, 2025. URL: <https://doi.org/10.31673/2412-4338.2025.011767>
2. Зінченко О. В., Звенігородський О. С., Кисіль Т. М., Згорткові нейронні мережі для вирішення задач комп'ютерного зору, 2022. URL: <https://doi.org/10.31673/2412-4338.2022.020411>
3. Мозговенко А. А., Зінов'єва О. Г., Аналіз методів комп'ютерного зору в задачах ідентифікації осіб у відеопотоці, 2022. URL: <https://doi.org/10.32851/tnv-tech.2022.3.6>
4. Синеглазов В.М., Прикладні системи штучного інтелекту: комп'ютерний зір, 2024. URL: <https://doi.org/10.15407/visn2024.06.043>
5. Тимошин Ю., Южда Ю., Аналіз особливостей застосування нейронних мереж для інтелектуальної обробки відеопотоків систем технічного зору, 2021. URL: <https://doi.org/10.20535/1560-8956.39.2021.247372>