Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та розроблення методів і наборів даних для розпізнавання осіб на відеоматеріалах низької роздільної здатності
(тема)

Виконав:
здобувач другого року навчання,
групи ДСм-24-1
Драконова О.О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Науки про дані (Data Science)
(повна назва спеціалізації)
Керівник доц. Волощук О.Б.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Лариса ЧАЛА
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Науки про дані (Data Science)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Драконовій Олесі Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розроблення методів і наборів даних для розпізнавання осіб на відеоматеріалах низької роздільної здатності

затверджена наказом університету від 24 листопада 20 25 р. № 1057Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 20 25 р.

3. Вихідні дані до роботи Офіційна документація з комп'ютерного зору, стандарти анотацій COCO, VisDrone, технічна документація фреймворку Ultralytics YOLO, бібліотек Python для машинного навчання, а також документація системи CVAT для розмітки зображень і відео, методичні матеріали кафедри штучного інтелекту щодо структури та вимог до кваліфікаційних робіт

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Дослідження об'єктів тестування та розробка методики проведення експерименту
3) Проєктування та тестування системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.12.2025	виконано
2	Визначення проблеми	25.11.2025	виконано
3	Аналіз конкурентних рішень	27.11.2025	виконано
4	Проектування компонентів системи	28.11.2025	виконано
5	Реалізація додатку	01.12.2025	виконано
6	Оформлення пояснювальної записки	06.12.2025	виконано
7	Захист перед ЕК	23.12.2025	виконано

Дата видачі завдання 24 листопада 20 25 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Волощук О.Б.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 59 с., 17 рис., 1 дод., 10 джерел.

АНАЛІЗ ЯКОСТІ ДАНИХ, АУГМЕНТАЦІЯ, ВІЙСЬКОВЕ ЗАСТОСУВАННЯ, ДАТА-СЕТИ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ, НИЗЬКА РОЗДІЛЬНА ЗДАТНІСТЬ, РОЗПІЗНАВАННЯ ЛЮДЕЙ НА ВІДЕО, ТРЕКІНГ, YOLO.

Об'єктом дослідження є сучасні методи та інструменти розпізнавання людей у відеопотоці низької роздільної здатності, зокрема в умовах, наближених до військових, де якість сигналу, шум та стиснення суттєво впливають на ефективність алгоритмів комп'ютерного зору.

Метою роботи є проведення комплексного аналізу існуючих дата-сетів для детекції людей, розробка інструментів їх уніфікації до формату COCO, створення системи тестування для порівняння моделей, а також формування рекомендацій щодо побудови спеціалізованих дата-сетів, адаптованих до задач розпізнавання в умовах деградованого відео.

Методами дослідження є використання згорткових нейронних мереж та архітектур сімейства YOLO для детекції, сучасних алгоритмів трекінгу для аналізу послідовності кадрів, а також розробка програмного інструментарію для автоматичного перетворення різних структур анотацій у формат COCO. Дослідження включає аугментацію та штучне погіршення відео з метою моделювання бойових умов, навчання моделей на різних датасетах та подальший аналіз їхньої ефективності на основі метрик точності та стійкості.

ABSTRACT

Master’s thesis contains: 59 pp., 17 fig., 1 ann., 10 references.

AUGMENTATION, COMPUTER VISION, DATASETS, DATA QUALITY ANALYSIS, NEURAL NETWORKS, LOW RESOLUTION, PEOPLE RECOGNITION IN VIDEO, TRACKING, MILITARY APPLICATIONS, YOLO.

The object of the research is modern methods and tools for recognizing people in low-resolution video streams, in particular in conditions close to military ones, where signal quality, noise and compression significantly affect the effectiveness of computer vision algorithms.

The aim of the work is to conduct a comprehensive analysis of existing datasets for human detection, develop tools for their unification to the COCO format, create a testing system for comparing models, and formulate recommendations for building specialized datasets adapted to recognition tasks in degraded video conditions.

The research methods are the use of convolutional neural networks and YOLO family architectures for detection, modern tracking algorithms for frame sequence analysis, and the development of software tools for automatic conversion of various annotation structures to the COCO format. The research includes video augmentation and artificial deterioration for the purpose of modeling combat conditions, training models on different datasets, and further analysis of their effectiveness based on accuracy and stability metrics.

ЗМІСТ

Вступ 7
1 Аналіз предметної галузі та постановка задачі 9
1.1 Сучасний стан технологій розпізнавання 9
1.1.1 Область розпізнавання людини 12
1.1.2 Аналіз методології та сфер застосування 15
1.2 Аналіз проблематики та існуючих рішень 22
1.3 Область проведення дослідження 24
1.4. Постановка задачі 26
2 Дослідження об'єктів тестування та розробка методики проведення експеременту 27
2.1 Вибір дата-сетів 27
2.2 Вибір інструментів тестування 30
2.3 Аналіз літературних та наукових джерел 32
2.4 Методика проведення дослідження 33
3 Проектування та тестування системи 35
3.1 Загальна архітектура 35
3.1.1 Модуль організації структури дата-сетів 36
3.1.2 Модуль навчання моделей YOLO та інтеграції з реальними даними 36
3.1.3 Модуль тестування моделей на зображеннях та аналізу результатів 37
3.2 Уніфікація дата-сетів 39
3.3 Тестування моделей 43
3.4 Проведення дослідження 45
3.5 Створення власного дата-сету 52
Висновки 56
Перелік джерел посилання 58
Додаток А Відомість кваліфікаційної роботи 59

ВСТУП

У сучасному світі технології комп’ютерного зору та штучного інтелекту відіграють ключову роль у сферах безпеки, оборони та автоматизації спостереження. Зі зростанням використання безпілотних літальних апаратів, мобільних камер та систем відеомоніторингу виникає потреба у надійних методах розпізнавання людей навіть у складних умовах. Особливо актуальним це стає в умовах низької роздільної здатності, шумів, спотворень та поганої якості сигналу, які характерні для військових та екстремальних сценаріїв. Розвиток алгоритмів детекції та трекінгу, зокрема моделей сімейства YOLO, відкриває можливості для автоматизації виявлення людей на будь-яких типах відеопотоків, однак ефективність цих методів у значній мірі залежить від якості та структури дата-сетів, на яких вони тренуються.

Одним із визначальних напрямів сучасних досліджень є аналіз та вдосконалення навчальних наборів даних, що використовуються для розпізнавання людей. Більшість популярних дата-сетів створені за стабільних, цивільних умов – із високою роздільністю, доброю освітленістю та мінімальною кількістю артефактів. Це істотно відрізняється від реальних умов застосування в бойових або критичних ситуаціях, де відео часто має сильну компресію, перешкоди, коливання камери, низький бітрейт та сильний шум. Саме тому дослідження, спрямоване на аналіз ефективності різних дата-сетів і формування рекомендацій щодо створення спеціалізованих наборів даних для розпізнавання людей у LQ-видео, є надзвичайно актуальним.

Завдяки розвитку методів глибинного навчання, зокрема згорткових нейронних мереж та моделей YOLO, стало можливим проводити детекцію людей у режимі реального часу навіть на ресурсно обмежених пристроях. Проте якість навчальних даних визначає межу точності, яку може досягти будь-яка модель. Тому дане дослідження передбачає не створення системи

детекції, а аналіз її базового елементу – дата-сетів, їх структури, якості та придатності до роботи з низькою роздільною здатністю, включаючи автоматизацію форматування та подальшу оцінку результатів навчання на кожному наборі.

Актуальність теми обумовлена потребою у стандартизованому підході до оцінки дата-сетів для задачі розпізнавання людей у несприятливих умовах. У межах роботи розглядаються особливості побудови сучасних навчальних наборів, способи їхньої уніфікації до формату COCO, методи аугментації та штучного погіршення якості, а також проведення експериментального порівняння результатів навчання моделі YOLO на різних наборах даних. Дослідження зосереджується на знаходженні закономірностей, які визначають ефективність дата-сету в умовах низькоякісного відеосигналу, та на формуванні методичних рекомендацій щодо створення майбутніх спеціалізованих датасетів військового призначення.

Мета даної роботи – вивчення теоретичних основ розпізнавання людей у відеопотоці, аналіз існуючих підходів до створення дата-сетів і розробка інструментарію для їхнього тестування, уніфікації та розробки власного адаптованого дата-сету.

Отримані результати стануть основою для подальшої розробки високоякісних наборів даних, які дозволять суттєво підвищити точність і стабільність алгоритмів розпізнавання людей у складних умовах, зокрема на відео низької роздільної здатності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Сучасний стан технологій розпізнавання

Розпізнавання зображень є одним із ключових напрямів комп'ютерного зору, що полягає у здатності алгоритмів автоматично інтерпретувати візуальну інформацію, подану у вигляді фотографій або відео. Його основне завдання – перетворити необроблені пікселі в семантично значущі об'єкти, такі як люди, транспорт, будівлі чи інші класи, що мають практичний сенс у конкретній задачі. На відміну від класичних методів комп'ютерної графіки, які лише оперують формами та кольорами, системи розпізнавання мають імітувати когнітивні можливості людини – визначати, що саме зображено на сцені, які об'єкти присутні та які між ними взаємозв'язки. Це перетворює розпізнавання зображень на фундаментальний інструмент сучасного штучного інтелекту.

Процес розпізнавання включає кілька основних етапів обробки. Першим є попередня підготовка даних – нормалізація зображень, шумозниження, підвищення контрасту та стандартизація розміру. Наступним етапом виступає виділення ознак: традиційні методи використовували SIFT, SURF, HOG і інші дескриптори, однак сучасний підхід базується на глибоких нейронних мережах, які самостійно навчаються витягувати високорівневі абстракції. Завершальним етапом є класифікація або детекція об'єктів в залежності від задачі система або визначає клас зображення загалом, або отримує координати об'єктів на сцені. Такий підхід дозволяє алгоритмам розпізнавати візуальні об'єкти з точністю, що часто перевищує людські можливості в складних умовах [2].

Розпізнавання зображень стало одним із базових компонентів у медицині, зокрема у задачах діагностики, де системи визначають аномалії на МРТ, рентгенівських або КТ-знімках. Завдяки цьому вдалося значно прискорити процес виявлення онкологічних або кардіологічних патологій,

зменшивши ризик лікарських помилок. У промисловості технологія застосовується для контролю якості продукції – автоматичні системи виявляють дефекти, відхилення від геометрії або порушення у структурі матеріалів. Такий підхід підвищує продуктивність виробництва і дозволяє знизити витрати на ручний контроль.

Важливою сферою застосування розпізнавання є транспорт та безпека дорожнього руху. Алгоритми виявляють дорожні знаки, аналізують поведінку автомобілів, виявляють пішоходів і прогнозують можливі зіткнення. Це є невід'ємною частиною систем автономного водіння, де обробка зображення камери відбувається в реальному часі, а рішення повинні прийматися з мінімальною затримкою. У поєднанні з Лідарами та радарами комп'ютерний зір формує повну картину навколишнього середовища, дозволяючи автомобілям безпечно переміщуватися.

У військовій сфері технології розпізнавання зображень відіграють критичну роль у розвідці, моніторингу та наведенні на ціль. Дрони, літаки та наземні системи використовують алгоритми для визначення техніки, особового складу, укріплень і підозрілих об'єктів. Застосування комп'ютерного зору дозволяє знизити ризики для особового складу та підвищити точність прийняття рішень, особливо в умовах, коли швидка обробка інформації визначає успіх операції. Автоматизовані системи спостереження здатні працювати цілодобово і виявляти загрози незалежно від погодних умов.

У цивільній безпеці розпізнавання зображень використовується у відеоспостереженні для виявлення нетипової поведінки, ідентифікації людей, визначення підозрілих об'єктів і запобігання потенційним загрозам. Системи аналітики перетворюють звичайні камери у потужний інструмент моніторингу, здатний автоматично відстежувати події та допомагати операторам своєчасно реагувати. Пошуково-рятувальні операції також активно покладаються на візуальне розпізнавання: дрони та роботи

використовуються для пошуку людей у важкодоступних або небезпечних зонах.

Окремий напрям розвитку – це розпізнавання облич, яке стало стандартом у смартфонах, банківських системах та технологіях контролю доступу. Алгоритми здатні аналізувати десятки мільйонів облич за секунди, зіставляючи їх із базами даних та визначаючи особу людини з високою точністю. Такі системи працюють навіть при зміні освітлення, віку або пози обличчя, що робить їх універсальними у реальному світі.

Широкого розвитку набули системи для класифікації діяльності або поведінки людей – наприклад, розпізнавання жестів, спортивних дій або нетривіальних патернів руху. Це використовується у спорті, фітнесі, реабілітації, а також у робототехніці, де машина повинна розуміти наміри людини. Аналіз руху, на відміну від статичного розпізнавання, вимагає обробки послідовностей кадрів, що накладає додаткові вимоги до продуктивності систем.

У цифрових сервісах – таких як соціальні мережі, онлайн-магазини чи рекомендаційні системи – розпізнавання зображень допомагає автоматично класифікувати контент, відбирати релевантні матеріали та підвищувати ефективність взаємодії користувача з платформою. Наприклад, алгоритми здатні пропонувати товари на основі фото, розпізнавати предмети в кадрі або автоматично покращувати якість зображень.

Загалом розпізнавання зображень стало універсальним інструментом сучасних цифрових технологій. Воно інтегроване в майже всі галузі – від медицини й промисловості до військової справи та побутових пристроїв. Подальший розвиток нейронних мереж, зокрема глибинного навчання та трансформерних архітектур, продовжує розширювати можливості розпізнавання, покращуючи точність, швидкість і здатність працювати у складних умовах. Це робить цю технологію однією з найважливіших складових сучасних інтелектуальних систем.

1.1.1 Область розпізнавання людини

Розпізнавання зображень як технологія охоплює широкий спектр завдань, спрямованих на автоматичну інтерпретацію візуальних даних, і забезпечує можливість виділяти об’єкти та їх характеристики. Проте серед усіх напрямів особливе місце займає розпізнавання людини – один із найважливіших і найскладніших підпроцесів комп’ютерного зору. Логічним продовженням загальної концепції розпізнавання зображень є перехід від абстрактних об’єктів до специфічної категорії – людей, оскільки саме вони мають ключову практичну цінність у задачах безпеки, аналітики, взаємодії «людина–машина» та автономних систем.

Розпізнавання людини включає широку групу методів, завдання яких – не лише виявити людину в кадрі, але й інтерпретувати її форму, положення, поведінку або ідентичність. У найпростішому вигляді це процес визначення присутності людини на зображенні або у відео. Однак у сучасних системах вимоги значно ширші: алгоритми повинні працювати при різних умовах освітлення, з різних ракурсів, у натовпі, при часткових перекриттях і навіть тоді, коли якість відео є низькою. Через це розпізнавання людини стало комплексною задачею, що включає детекцію, сегментацію, трекінг, оцінку поз (pose estimation) та ідентифікацію.

Однією з найбільш поширених задач є детекція людини, яка передбачає визначення координат її розташування в кадрі. Тут широко застосовуються згорткові нейронні мережі (Faster R-CNN, YOLO, SSD) і трансформерні архітектури (DETR), що здатні виявляти людину навіть у складних сценах. На відміну від інших об’єктів, людина має високу варіативність зовнішнього вигляду – одяг, положення тіла, аксесуари – що робить цю задачу значно складнішою, а особливо у відео низької роздільності, де дрібні деталі втрачаються [3].

Не менш важливою задачею є сегментація людини, яка визначає точну форму тіла, виділяючи пікселі, що належать людині, від фону. Це

використовується у відеоаналітиці, AR/VR-системах, робототехніці та медичній діагностиці. Сегментація дозволяє окремо виділяти силуети, що є критично важливим у випадках, коли точність контуру має значення – наприклад, для аналізу рухів, створення 3D-моделей або виявлення аномальних поз.

Окремий напрям – ідентифікація або розпізнавання особи, яка спрямована на визначення конкретної людини. Це використовує біометричні особливості, зокрема риси обличчя, структуру тіла, ходу (gait recognition). Технології ідентифікації використовуються у безпекових системах, контролі доступу, банківських сервісах, правоохоронній діяльності та персоналізованих цифрових платформах. Особливої уваги тут потребують етичні й правові аспекти, пов’язані з приватністю.

Значного розвитку набуло оцінювання поз людини (human pose estimation) – визначення положення ключових точок тіла, таких як лікті, коліна, кисті та інші суглоби. Це дозволяє інтерпретувати дії людини, аналізувати її поведінку, виявляти небезпечні ситуації або некоректні рухи. Системи оцінки поз широко використовуються у спорті, медицині, фізичній реабілітації, системах відеоспостереження та в аналітиці масових заходів.

Ще один напрям – аналіз поведінки людини, який вимагає особливої складності, оскільки він пов’язаний не лише з простим виявленням, а й з інтерпретацією дій у контексті. Такі системи визначають, чи людина рухається підозріло, чи падає, чи бере участь у небезпечній активності. У сфері безпеки це дозволяє автоматично виявляти інциденти, а у транспорті – забезпечувати контроль за пішоходами в режимі реального часу.

Розпізнавання людини активно застосовується в інтерактивних системах, де важливою є взаємодія з користувачем. Робототехніка, ігрова індустрія, gesture-controlled інтерфейси покладаються на точне визначення положення людини, її рухів і жестів. У таких системах точність розпізнавання прямо впливає на комфорт і якість взаємодії, а низька роздільність відео може суттєво погіршувати користувацький досвід.

У сфері безпеки та моніторингу розпізнавання людини з відео є ключовим елементом для автоматичного стеження, пошуку зниклих осіб, контролю за поведінкою натовпу та виявлення загроз. В умовах поганої якості відео – що характерно для дешевих камер, камер у темряві чи старих систем – задача значно ускладнюється. Тут особливого значення набувають алгоритми, здатні працювати в умовах низької чіткості та шумності, що є безпосереднім об’єктом дослідження цієї роботи.

Загалом розпізнавання людини є однією з фундаментальних складових сучасних інтелектуальних систем, оскільки людина є основним об’єктом уваги у більшості реальних сценаріїв. Від автономних машин до медичних систем, від спостереження до персоналізованих сервісів – можливість точно і швидко виявляти та інтерпретувати людину формує основу для безпечних, розумних і ефективних технологічних рішень. Особливої актуальності набуває проблема розпізнавання людини у відео низької роздільності, де класичні алгоритми стикаються з суттєвими обмеженнями, що потребують нових методів, підходів і спеціально підготовлених дата-сетів. На рисунку 1.1 наведено приклад детекції за допомогою YOLO 8.

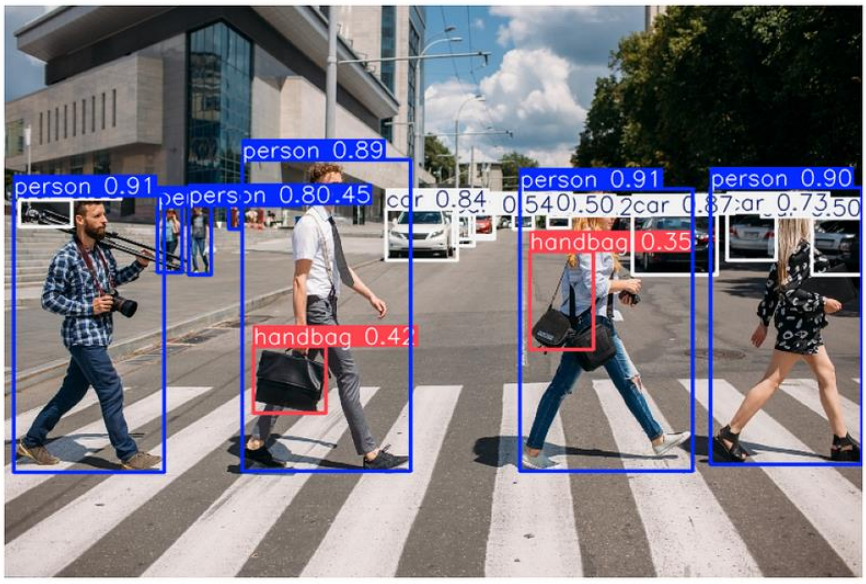

Рисунок 1.1 – Приклад виділення об’єктів декількох класів

На рисунку видно приклад роботи інтерфейсу YOLO 8 який виявив 3 типи об’єктів в тому числі і індексовану людину.

1.1.2 Аналіз методології та сфер застосування

Розпізнавання людини сьогодні використовується у широкому спектрі галузей, серед яких можна виокремити системи безпеки та відеоспостереження, медичну діагностику та моніторинг стану пацієнтів, інтерактивні та робототехнічні системи, транспорт та автономне водіння, спортивну аналітику, контроль доступу та біометричну ідентифікацію, а також військові та оборонні технології. Незважаючи на різноманітність застосувань, у сьогоднішніх реаліях – саме військова сфера є більш актуальною, яка потребує най масовішого застосування засобів розпізнавання та висуває найвищі вимоги до точності, швидкості та стійкості алгоритмів, особливо у випадках, коли якість відео є низькою, а умови – складними та непередбачуваними, тож розпізнавання допомагає знаходити наступні цілі:

– поранені, для евакуації;

– теплові сигнатури;

– сигнатури вогню чи диму;

– різноманітні цілі для ураження.

Але саме у військових застосуваннях розпізнавання людини має критичне значення, оскільки здатність швидко визначити присутність особового складу противника, його дії або стан напряму впливає на успіх операцій та збереження життя власного персоналу. Алгоритми комп’ютерного зору використовуються для виявлення сигнатур цілей – наприклад, характерних контурів людської фігури, навіть коли вона частково прихована, замаскована або знаходиться на значній відстані. Крім того, сучасні дрони, наземні платформи та системи розвідки повинні вміти розрізняти бойові одиниці, цивільних осіб та об’єкти, які не становлять

загрози. Особливо важлива можливість працювати з низькоякісним або перешкодженим відео, де деталі людського силуету спотворені шумом, поганим освітленням чи стисненням сигналу. Саме тому ефективність алгоритмів у LQ-відео має велике стратегічне значення, адже більшість польових камер, тепловізорів і FPV-дронів передають картинку з обмеженою роздільністю [4].

Ще один аспект – виявлення поранених чи непритомних військових. Алгоритми повинні розуміти нетипові пози тіла, детектувати людей, які не рухаються або лежать на землі, що критично для пошуково-рятувальних операцій і швидкої евакуації. Також системи аналізу відео допомагають визначати потенційно небезпечну поведінку, наприклад підготовку до атаки, переміщення груп противника або спроби проникнення на контрольовану територію. У контексті сучасних бойових дій, де широко використовуються FPV-дрони, наземні роботи та автономні турелі, здатність точно виявляти людину в різних умовах – у диму, у тепловізорі, при русі або в статичній позі – стає фактором, що визначає ефективність і безпеку застосування таких систем. Таким чином, розпізнавання людини у військовій сфері є не просто корисною технологією, а одним із ключових елементів сучасної бойової аналітики й високоточної навігації та ураження.

У військовій сфері для розпізнавання людини використовується широкий спектр технічних засобів, кожен з яких має власні особливості, переваги та обмеження. Одними з наймасовіших і найдинамічніших платформ є FPV-дрони, які використовуються для спостереження, наведення та ураження. Завдяки своїй маневровості та можливості працювати на малих висотах вони забезпечують безпосередній огляд поля бою, однак мають суттєві технічні обмеження – низьку роздільність відеопередачі, високий рівень шумів та стиснення сигналу, що ускладнює точне розпізнавання людей. Незважаючи на це, FPV-дрони є стратегічно важливими, оскільки дозволяють швидко виявляти переміщення

противника, ідентифікувати відкриті позиції та в реальному часі реагувати на тактичні зміни.

Другим типом засобів є розвідувальні дрони, які можуть бути як тактичними (малі та середні квадрокоптери), так і оперативно-тактичними (літакового типу з великою дальністю польоту). Такі платформи оснащуються високоякісними камерами денного бачення, тепловізорами, мультиспектральними сенсорами, що дозволяє здійснювати детекцію людини на великих відстанях і в складних умовах: вночі, під час туману, при густій рослинності чи в умовах задимлення. Розвідувальні дрони здатні передавати стабільніше та якісніше відео, порівняно з FPV, тому їх активно використовують для аналітики, оцінки позицій, корекції артилерійського вогню та пошуку рухомих груп супротивника.

Також широко застосовуються стаціонарні та мобільні комплекси відеоспостереження, встановлені на фортифікаційних позиціях, блокпостах або в охоронних периметрах. Такі системи можуть включати як звичайні камери, так і тепловізійні та інфрачервоні сенсори. Завдяки тривалому безперервному моніторингу вони дозволяють автоматично виявляти підхід ворожих диверсійних груп, фіксувати переміщення на великих територіях та інтегрувати дані у загальну систему ситуаційної обізнаності.

Не менш важливим засобом є наземні роботизовані системи, які виконують функції спостереження або ведення бою і оснащені різними типами оптики. Вони дозволяють вести розвідку в небезпечних зонах, де присутня загроза для особового складу, наприклад у міських руїнах або під час штурмових операцій. Надсучасні моделі інтегрують алгоритми штучного інтелекту безпосередньо на борту, що дає можливість здійснювати первинну обробку відео і розпізнавати людини без доступу до потужної хмарної інфраструктури.

Додатково у військових операціях використовуються тепловізори, прилади нічного бачення та мультиспектральні камери, які забезпечують можливість виявляти людей за ознаками теплового випромінювання або

відбиття світла різних спектральних діапазонів. Такі пристрої дають змогу здійснювати розпізнавання навіть у повній темряві або під сильними атмосферними перешкодами. Завдяки поєднанню цих засобів створюються комплексні системи ситуаційної обізнаності, де алгоритми аналізують різні типи сигналів та підвищують ймовірність виявлення людини.

Таким чином, сучасні військові технології використовують цілу екосистему засобів для розпізнавання людини – від FPV-дронів із низькою якістю відео до високоточних розвідувальних платформ і роботизованих комплексів. Ефективність таких систем значною мірою залежить від здатності алгоритмів комп'ютерного зору працювати в умовах обмежених ресурсів, шумних сигналів та низької роздільності, що робить цю область критично важливою для підвищення боєздатності сучасних армій.

Далі варто розглянути програмні компоненти які забезпечують увесь цикл розробки та реалізації процесу розпізнавання.

Програмні компоненти включають у себе дата-сети для навчання, нейронні мережі для детекції об'єктів (такі як YOLO) та трекінгові механізми складають фундамент сучасних систем розпізнавання людини. Вони виконують різні функції, але разом формують цілісний інтелектуальний ланцюг, який дозволяє машині бачити, розуміти та відстежувати людину у відео.

Дата-сет – це спеціально підготовлена колекція зображень або відео, які використовуються для навчання моделей комп'ютерного зору. Кожен елемент у дата-сеті має розмітку – інформацію, яка вказує моделі, що саме зображено на знімку. Для задач розпізнавання людини такі дата-сети містять:

– координати прямокутників (bounding boxes), де знаходиться людина;

– маски сегментації (точні контури силуету);

– ключові точки тіла (для визначення поз);

– ідентифікатори людей (для Re-ID);

– послідовності кадрів для трекінгу.

Чим більше різноманіття у дата-сеті (різні ракурси, освітлення, роздільність, одяг, перекриття), тим кращою буде модель.

Для військових цілей важливо мати дата-сети низької роздільності, з шумами, з камерами FPV та тепловізорами, оскільки моделі, треновані на HD-зображеннях, погано працюють у реальних бойових умовах.

Окремо розглянемо нейронні мережі типу YOLO. YOLO (You Only Look Once) – це сімейство нейронних мереж для детекції об’єктів у реальному часі. Їхня особливість у тому, що вони обробляють зображення за один проход, прогнозуючи:

– чи є на зображенні людина;

– де саме вона знаходиться (координати боксу);

– наскільки впевнена модель у цьому;

– інші об’єкти на сцені.

YOLO стала стандартом у системах, де важлива швидкість, наприклад:

– FPV-дрони;

– робототехніка;

– відеоспостереження;

– мобільні пристрої.

Основні версії: YOLOv5, YOLOv7, YOLOv8, YOLO-NAS, YOLOv10 – вони покращують точність, швидкість та ефективність.

У бойових умовах YOLO часто донавчають на спеціальних дата-сетах низької якості, щоб модель могла розпізнавати людей, які займають 10–20 пікселів у кадрі.

Механізми трекінгу – відіграють ключову роль у системах розпізнавання людини, оскільки саме вони забезпечують безперервне відстеження об’єкта у відеопотоці. Трекінг – це процес визначення того, чи є об’єкт, виявлений на поточному кадрі, тим самим, що й на попередніх. На

відміну від детектора, який працює з окремими кадрами незалежно, трекер створює часовий зв’язок між ними, додаючи системі пам’ять та контекст.

Робота трекера базується на інформації, яка надходить від детектора, наприклад YOLO. Трекінгові алгоритми використовують одразу кілька факторів для визначення відповідності: попередні координати об’єкта, його швидкість і напрямок руху, а також подібність зовнішнього вигляду між кадрами. Завдяки цьому система може знайти об’єкт, навіть якщо він трохи змістився, частково зник із кадру або змінив позу. Сучасні трекери поєднують як класичні методи прогнозування траєкторії, так і глибинні нейронні моделі для точнішого зіставлення об’єктів.

Серед найпопулярніших трекінгових алгоритмів виділяється DeepSORT, який використовує нейронні ознаки зовнішності людини для точного зіставлення в різних кадрах. BYTETrack демонструє високу ефективність при низькій якості відео й здатний утримувати об’єкт навіть за умов часткових пропусків детекцій. OC-SORT забезпечує стійкість до шумів і стрибків кадрів, тоді як BoT-SORT добре справляється зі складними сценами, де люди перекривають одна одну або рухаються в густих групах. Кожен із цих трекерів оптимізований для роботи у своїх специфічних умовах, але всі вони виконують одну спільну задачу – зберігати тяглість спостереження.

Діаграма (рисунок 1.2) відображає повний цикл взаємодії між дата-сетом та згортковою нейронною мережею (CNN), демонструючи, як сирі зображення та відео з розміткою проходять через етапи попередньої обробки, аугментації та подачі у DataLoader, який формує навчальні батчі для моделіх [5].

Під час навчання Training Loop оновлює ваги CNN, після чого модель може виконувати інференс і видавати результати детекції у вигляді координат bounding boxes, масок, ключових точок чи оцінок упевненості. Діаграма також показує зворотний зв’язок – результати можуть

використовуватися для донабору або уточнення дата-сету, що забезпечує безперервне вдосконалення моделі.

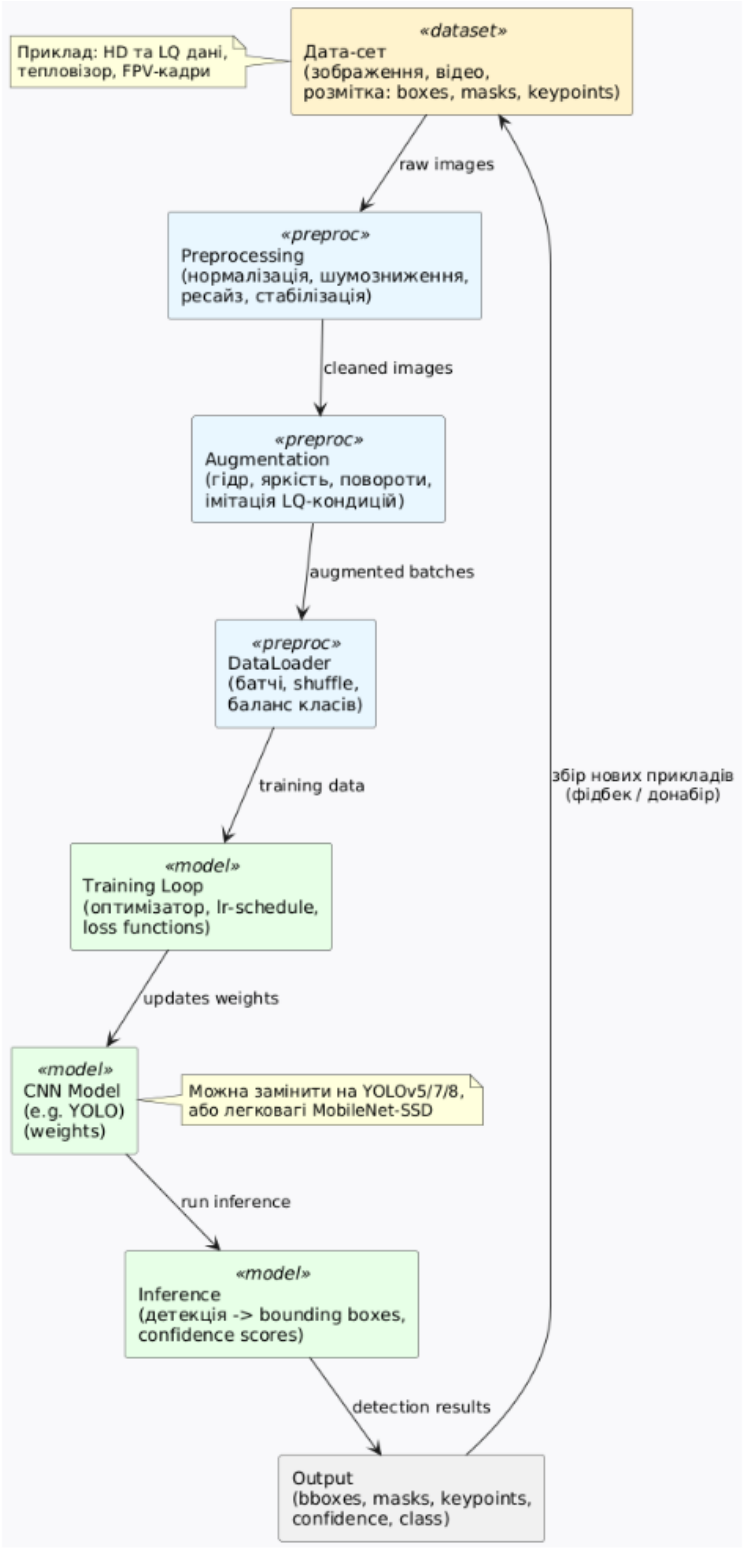

Рисунок 1.2 – Діаграми зв’язку між дата-сетом та CNN

1.2 Аналіз проблематики та існуючих рішень

Однією з ключових проблем у системах військового розпізнавання людини є складність вибору правильних інструментів та алгоритмів, оскільки умови, у яких працює оптичне обладнання, суттєво відрізняються від лабораторних чи цивільних сценаріїв. Камери, встановлені на FPV-дронах, розвідувальних платформах, наземних роботах або стаціонарних постах спостереження, зазвичай мають низьку або обмежену роздільну здатність, вузьку смугу пропускання відеосигналу та нестабільний зв’язок.

У реальних бойових умовах відео часто передається із сильним стисненням, артефактами, втратами кадрів, затримками або перешкодами, що суттєво знижує якість зображення і створює додаткові виклики для алгоритмів комп’ютерного зору.

Ці технічні обмеження породжують проблему адаптації сучасних моделей розпізнавання, оскільки більшість з них навчається на високоякісних датасетах з чіткими та деталізованими зображеннями. У бойових відеопотоках людина може виглядати як кілька десятків пікселів, мати розмиті контури, бути частково перекритою або зовсім невпізнаваною для алгоритмів, які очікують наявність чітких візуальних ознак.

Через це традиційні моделі часто демонструють низьку точність, високу кількість хибних спрацювань або взагалі не здатні стабільно розпізнавати людей у складних умовах.

На рисунку 1.3 видно погану роздільну здатність та артефакти які є наслідком поганої якості камери та шуму у сигналі відеопотоку. Це все ускладнює виклики покладені на механізми розпізнавання.

Додатково слід враховувати, що кожне бойове завдання накладає власні вимоги до швидкості обробки, оскільки рішення потрібно приймати в режимі реального часу. FPV-дрон, що летить на швидкості 100–150 км/год, потребує аналізу кожного кадру з мінімальною затримкою, щоб виявити людину, техніку чи інші важливі об’єкти. Це означає, що навіть дуже точні

алгоритми, які потребують надмірних обчислювальних ресурсів, стають непрактичними для польового застосування.

Рисунок 1.3 – Приклад стану відеосигналу в бойових умовах

Більшість сучасних дата-сетів для комп’ютерного зору створені під цивільні задачі та високоякісні умови зйомки, тому вони не відповідають вимогам військового застосування. Типові набори даних містять чіткі, добре освітлені зображення з високою роздільністю, стабільною камерою та мінімальною кількістю артефактів. У таких наборах деталі людської фігури добре помітні, контури не спотворені, а колірна гамма передається коректно.

Для моделей це означає, що вони навчаються розпізнавати людей за чіткими й виразними ознаками, які майже ніколи не зустрічаються у реальних бойових умовах. У результаті, алгоритми, треновані на стандартних дата-сетах, різко втрачають точність під час роботи з низькоякісним відео, оскільки не мають досвіду роботи з такими «слабкими» сигналами.

У військовому середовищі джерела відео є суттєво гіршими: FPV-дрони, малопотужні розвідувальні коптери та тепловізори працюють із сильним стисненням, шумами, втратою кадрів, завадами та низькою роздільністю. Людина може займати лише кілька десятків пікселів, бути розмитою, частково прихованою або знятою під екстремальними кутами. Присутні артефакти компресії, смуги перешкод, зернистість та викривлення сигналу через радіозавади. Такі умови принципово відрізняються від тих, що закладені у класичних дата-сетах, і тому сучасні моделі не мають достатньої стійкості до шумних і нестабільних бойових відеопотоків. Це створює гостру потребу у формуванні спеціалізованих військово-орієнтованих дата-сетів, адаптованих для роботи з реальними, а не лабораторними даними.

Таким чином, проблема вибору оптимальних інструментів для розпізнавання у військових умовах полягає не лише в якості моделей, а й у необхідності адаптувати їх до низькоякісного відео, нестабільного сигналу та обмежених ресурсів обчислення. Це створює потребу порівнянні існуючих дата-сетів, аналізі ефективності чи не ефективності підходів які в них застосовані та розробці паттерну для оптимальних даних для навчання нейронних мереж.

1.3 Область проведення дослідження

Область проведення дослідження стосується комп’ютерного зору, а саме – задачі розпізнавання людини у відеопотоці з низькою роздільною здатністю, що є одним із найбільш вимогливих і динамічних напрямів сучасних технологій штучного інтелекту. У центрі цієї галузі лежить здатність алгоритмів аналізувати візуальні сигнали, виділяти людей серед складного фону, інтерпретувати їхню поведінку та підтримувати стійкість роботи в умовах, далеких від ідеальних. Наукові дослідження в цій сфері поєднують у собі аспекти глибинного навчання, комп’ютерної графіки,

математичного моделювання та обробки сигналів, що робить її міждисциплінарною та технічно насиченою.

Особливого значення даний напрям набуває у випадках, коли джерелом інформації є відео низької якості: FPV-потоки з дронів, тепловізори, дешеві камери спостереження або системи, що працюють в умовах обмеженої пропускної здатності. Саме такі відеодані містять різноманітні перешкоди – шуми, артефакти компресії, розмиття через рух, тремтіння камери, часткові перекриття та нестабільну освітленість. Для традиційних моделей розпізнавання це створює низку труднощів, оскільки вони зазвичай розроблені та навчені на високоякісних дата-сетах, які не відображають реалій несприятливих сценаріїв. Тому ключовим завданням є адаптація існуючих підходів, розробка нових архітектур і створення спеціалізованих наборів даних.

У сучасних дослідженнях значну увагу приділяють аналізу алгоритмів детекції та трекінгу, а також підбору оптимальних методів, що зберігають точність навіть при деградації відеосигналу. Особлива роль належить нейронним мережам типу YOLO, які відзначаються високою швидкістю роботи, та трекерам, що дозволяють об’єднати детекції у стабільні траєкторії. Досліджується питання того, як комбінувати ці модулі, адаптувати їхні гіперпараметри, використовувати попередню обробку та аугментацію для кращої стійкості до шуму та втрат. Також важливим напрямом є створення дата-сетів, що моделюють реальні умови бойових дій або інших складних сценаріїв.

Таким чином, область проведення дослідження об’єднує питання побудови дата-сетів, аналізу сучасних моделей комп’ютерного зору та розробки стійких методів розпізнавання людини у низькоякісному відео. Це дозволяє сформувати наукове підґрунтя для вирішення практичних задач, пов’язаних із безпекою, військовими технологіями, пошуково-рятувальними операціями та інтелектуальними системами спостереження.

1.4. Постановка задачі

Метою дослідження буде аналіз ефективності існуючих дата-сетів для задачі розпізнавання людей на відео низької роздільної здатності та розробка інструментарію, який дозволить уніфікувати, тестувати та порівнювати такі набори даних у стандартизованих умовах. Дослідження спрямоване не на створення повної системи детекції, а на формування науково обґрунтованої основи для створення спеціалізованих дата-сетів, адаптованих до умов військового застосування – з шумами, деградованою якістю та нестабільним відеопотоком. Основними завданнями є відбір найбільш поширених датасетів для виявлення людей, їх приведення до формату COCO, проведення навчання моделі YOLO на кожному з них, тестування результатів у нормальних та деградованих умовах, а також аналіз отриманих відмінностей для виявлення характеристик, які впливають на точність розпізнавання.

Результатом роботи стануть інструменти та аналітичні матеріали, які:

– забезпечують автоматизоване перетворення сторонніх дата-сетів у формат COCO для подальшого використання в алгоритмах детекції;

– створюють єдину тестову платформу для навчання та порівняння моделей на різних наборах даних, включаючи можливість генерувати їх деградовані (LQ) варіанти;

– дозволяють виконати структурний та статистичний аналіз ефективності кожного датасету під час навчання YOLO;

– формують рекомендації щодо створення спеціалізованих дата-сетів для відео низької роздільної здатності, визначаючи оптимальні властивості анотацій, структури кадрів, балансу сцен, рівня шумів та варіативності об’єктів;

– закладають основу для подальшої розробки власного спеціального дата-сету, який буде максимально адаптований до військових умов та задач розпізнавання людей у реальних польових сценаріях.

2 ДОСЛІДЖЕННЯ ОБ'ЄКТІВ ТЕСТУВАННЯ ТА РОЗРОБКА МЕТОДИКИ ПРОВЕДЕННЯ ЕКСПЕРЕМЕНТУ

2.1 Вибір дата-сетів

У системах комп’ютерного зору формати дата сетів відіграють ключову роль, оскільки визначають спосіб зберігання зображень, структуру метаданих та організацію розмітки, необхідної для навчання моделей глибокого навчання. Найбільш поширені підходи включають використання наборів файлів зображень у поєднанні з текстовими або JSON-файлами, що містять координати об’єктів та їх класові мітки. Стандартизовані формати дозволяють забезпечити узгодженість між різними моделями та інструментами анотації, полегшують передачу даних між дослідженнями та створюють можливість масштабування навчальних систем без необхідності ручного переформатування великих масивів даних.

Багато сучасних наборів даних розроблені таким чином, щоб бути максимально універсальними: вони включають різні типи розмітки – від класифікації до сегментації та виявлення об’єктів. Це дозволяє датасетам бути сумісними з широким спектром алгоритмів, зокрема детекторами, сегментаторами та рекогнайзерами. Правильна організація формату – критичний аспект, особливо у задачах з великою кількістю об’єктів чи обмеженою якістю відео, оскільки некоректна структура анотацій може призвести до суттєвої втрати якості під час тренування моделі.

Одним із найпопулярніших форматів структурування даних для задачі детекції об’єктів є формат, що використовується у моделях сімейства YOLO. Він зосереджений на максимальній простоті та швидкодії, дозволяючи моделі читати анотації без надмірних метаданих. Кожне зображення має відповідний текстовий .txt файл, який містить координати всіх об’єктів у нормалізованій формі. Такий підхід значно прискорює

обробку даних і робить формат ідеальним для реального часу, адже YOLO не витрачає час на парсинг складних структур, як у COCO чи Pascal VOC[6].

YOLO не зберігає додаткових атрибутів, таких як ім’я файлу, загальний розмір чи метадані камери – все це передбачається і так зчитується з окремого зображення. Таким чином, анотації перетворюються на компактний і універсальний формат, який може бути адаптований під різні типи об'єктів, включаючи людей, техніку та інші цілі, що зустрічаються у реальних сценаріях.

Файл анотації для одного зображення містить рядки такого формату:

– class_id – числовий індекс класу об’єкта (0, 1, 2…);

– x_center – нормалізована координата центра об'єкта по осі X (від 0 до 1);

– y_center – нормалізована координата центра по осі Y (від 0 до 1);

– width – нормалізована ширина bounding box (від 0 до 1);

– height – нормалізована висота bounding box (від 0 до 1).

Нижче на рисунку 2.1 наведено приклад подібної анотації на три класи.

```
0 0.525773 0.7825 0.149953 0.1925
0 0.812559 0.62 0.179944 0.2575
1 0.628866 0.3125 0.263355 0.32625
```

Рисунок 2.1 – Приклад структури анотацій YOLO

Поряд із широко застосовуваними форматами розмітки, такими як COCO та YOLO, існують доменно-орієнтовані підходи, розроблені для більш специфічних задач. Використання одразу двох різних форматів у дослідженні дає можливість глибше оцінити універсальність алгоритмів стандартизації анотацій та побудувати методику перетворення, придатну для подальшої автоматизації. Саме з цієї причини до аналізу включено формат VisDrone, який є другим за популярністю у сфері повітряного

моніторингу, містить відмінну структуру анотацій та суттєво відрізняється від класичних форматів типу COCO.

Формат VisDrone створено спеціально для завдань детекції об’єктів на зображеннях БПЛА, де характерними є значні зміни ракурсу, масштабу, щільності об’єктів і варіативність освітлення. Подібні умови змушують формати розмітки бути більш деталізованими та містити додаткові службові параметри, які відсутні в універсальних наборах даних. Важливою особливістю VisDrone є використання структурованих текстових файлів, у яких об’єкти описуються не лише через координати прямокутника, але й через інформацію про стан об’єкта та якість його анотації.

У дата-сеті VisDrone кожне зображення супроводжується текстовим файлом, де кожний рядок відповідає окремому об’єкту, а всі поля мають фіксовані позиції. На відміну від YOLO чи COCO, тут використовуються абсолютні координати (x, y, width, height), починаючи від верхнього лівого кута зображення. Також додаються службові поля, що дозволяють фільтрувати об’єкти під час навчання, наприклад, за ступенем оклюзії або видимістю. Це робить формат більш насиченим та інформативним, але водночас менш універсальним і складнішим у використанні без попереднього перетворення [7].

Структура розмітки VisDrone (один рядок = один об'єкт):

– x – координата лівого верхнього кута області детекції;

– y – координата верхнього краю області детекції;

– w – ширина bbox у пікселях;

– h – висота bbox у пікселях;

– category_id – ID класу об'єкта (1–10);

– truncation – ступінь обрізання об’єкта (0–1);

– occlusion – ступінь закриття об’єкта іншими об’єктами (0–2);

– ignored – ознака, чи слід ігнорувати об'єкт під час тренування.

Нижче на рисунку 2.2 наведено приклад подібної анотації на три класи.

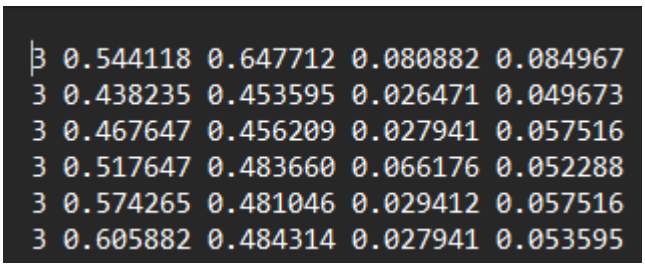

Рисунок 2.2 – Приклад структури анотацій VisDrone

2.2 Вибір інструментів тестування

Було використано наступні інструменти:

– Python;

– бібліотека Ultralitycs;

– OpenCV;

– Visual Studio Code;

– моделі YOLO;

– CUDA.

Python використовується як базове середовище для реалізації обробки даних, підготовки анотацій, навчання моделей та тестування результатів. Мова має велику екосистему бібліотек, орієнтованих на машинне навчання, комп'ютерний зір та обробку зображень, включаючи NumPy, OpenCV, PyTorch та інші. Застосування Python дає можливість створювати компактний, читабельний та легко модифікований код для автоматизації всіх етапів роботи – від стандарталізації датасетів до запуску моделей на відео.

Ultralytics – це офіційна екосистема, що включає реалізацію сучасних моделей YOLO, інструменти тренування, тестування, валідації та експорту моделей. Вона забезпечує високий рівень автоматизації процесів, завдяки чому можливо швидко запускати тренування, переглядати прогрес, формувати структуру проєкту та виконувати детекцію на нових даних. Використання Ultralytics важливе для дослідження, оскільки воно надає

стабільну реалізацію YOLO з підтримкою різних архітектур, наборів параметрів та сучасних GPU.

Моделі сімейства YOLO обрано як основні інструменти детекції об'єктів завдяки їхній високій швидкості, точності та оптимізації для реального часу. Формат навчання, анотацій та інтерфейс моделі добре документовані, що значно спрощує інтеграцію. У роботі застосовуються готові архітектури (наприклад, YOLO11), які дозволяють тренувати модель на користувацьких датасетах, таких як VisDrone, і отримувати готові ваги для подальшого використання у тестових сценаріях.

VS Code обрано як основний інструмент розробки через його гнучкість, підтримку Python, інтегровані термінали та можливість підключення розширень для роботи з Git, Python, документацією та форматуванням коду. Середовище дозволяє ефективно організовувати структуру проєкту, керувати файлами датасетів, запускати скрипти тренування та тестування в одному робочому просторі [8].

CUDA використовується для прискорення навчання та тестування моделей завдяки перенесенню обчислень з процесора на графічний адаптер NVIDIA. GPU-обчислення дають можливість суттєво зменшити час тренування, особливо при роботі з великими датасетами, такими як VisDrone. Інтеграція з CUDA є критичною частиною процесу, оскільки дозволяє працювати з сучасними моделями YOLO у повноцінному продуктивному режимі, забезпечуючи обробку відеопотоків у реальному часі.

OpenCV застосовується як інструмент для зчитування відеопотоків, обробки кадрів та візуалізації результатів детекції. Саме завдяки OpenCV можливо реалізувати компонент тестування, який накладає рамки, підписи класів і процедуру обробки кадрів у циклі. Це є необхідною частиною системи, оскільки дозволяє оцінити роботу моделі на реальних відеоданих і провести аналіз швидкодії та стабільності.

2.3 Аналіз літературних та наукових джерел

Далі варто розглянути наукові праці, що вивчають особливості побудови дата-сетів і методи розпізнавання людини, зокрема в умовах змінної якості відео. Обрані джерела представляють різні підходи - від систематичного оцінювання наявних наборів даних до використання комбінованих моделей, що враховують як візуальні, так і рухові характеристики. Аналіз таких робіт дозволить виявити ключові тенденції, зрозуміти обмеження існуючих наборів даних і визначити напрямки удосконалення, які є релевантними для задач розпізнавання у низькоякісних відеопотоках.

A Systematic Evaluation and Benchmark for Person Re-Identification: Features, Metrics, and Datasets (S. Karanam, M. Gou, Z. Wu та ін.).

Ця робота є фундаментальним оглядом і бенчмарком для систем ре-ідентифікації людей (person re-ID), що охоплює як алгоритми, так і набори даних. Автори проаналізували 11 різних методів екстракції ознак та 22 підходи до метрик зіставлення, протестувавши їх на багатьох публічних re-ID дата-сетах (наприклад, VIPeR, DukeMTMC, Market-1501 та ін.), для створення єдиного порівняльного середовища.

Цей огляд корисний для нашої теми тим, що показує, які саме характеристики (features) та метрики краще підходять для розпізнавання людей у реалістичних сценаріях спостереження. Оскільки наш фокус – відео низької роздільної здатності (low-res), важливо зрозуміти, які набори даних та методи re-ID вже використовуються, їх сильні й слабкі сторони, щоб оцінити, як їх можна модифікувати або доповнити новим дата-сетом, більш релевантним до військових умов.

Video-based Person Re-identification with Accumulative Motion Context (AMOC) (H. Liu, Z. Jie, J. Karlekar та ін.).

У цій статті запропонована мережа AMOC, яка враховує не лише зовнішній вигляд особи (appearance), а й контекст руху (motion context) із

відеопослідовностей. Мережа має двопоточну архітектуру: один потік обробляє ознаки зовнішнього вигляду, а другий – рухову інформацію з суміжних кадрів. Потім вони комбінуються через рекурентний модуль, що дозволяє акумулювати довготривалі сигнали руху.

Експерименти виконувалися на відео-ре-ID датасетах (iLIDS-VID, PRID-2011, MARS), і результати показали, що AMOC значно перевершує класичні підходи, особливо в умовах часткових перекриттів, зміни пози чи поганої якості кадрів.

Це дослідження має пряму релевантність для військового контексту: відео з дронів або спостережних систем часто містять шум, часткові перекриття, низьку чіткість і динаміку рухів. Підхід AMOC показує, що об’єднання рухової інформації може значно підвищити надійність розпізнавання людини в таких складних умовах. Для нашого дослідження це дає основу для розгляду віде-ре-ID алгоритмів або розробки модифікованих мереж, які краще працюватимуть на низькоякісних відео.

2.4 Методика проведення дослідження

Методика дослідження буде спрямована на систематичну перевірку впливу різних існуючих дата-сетів на якість роботи детекторів у умовах низької роздільної здатності та шумного відео, а також на виведення шаблонів (патернів) тих наборів даних, які дають найкращий результат. Далі варто розглянути покроковий опис стратегії, експериментального конвеєра та методів аналізу:

– для початку необхідно буде зробити відбір дата-сетів. ми оберемо кілька широко використовуваних та різнохарактерних публічних наборів для задач виявлення і ре-id людей (наприклад, coco/person, pascal voc (person subset), crowdhuman, market-1501/mars для відео/ре-id, тепловізійні або спеціалізовані датасети якщо доступні). важливі критерії відбору – різноманітність сцен (натовп/розріджений простір), роздільна здатність

джерел, наявність сегментаційних/ключових точок та якість анотацій. Мета – отримати сукупність датасетів, які відображають різні аспекти реальних умов;

– етап уніфікація формату. для порівнянності всі обрані набори будуть перетворені у єдиний формат coco (json з полями images/annotations/categories).

– імітація умов низької якості (data augmentation / corruption). оскільки ціль – бойові/польові відеопотоки, буде підготовлено відеопотік низької якості для тестування;

– навчання CNN. буде вибрано одну (або кілька) репрезентативних архітектур YOLO (наприклад, yolov5-small, yolov8-nano) та зафіксовано базовий набір гіперпараметрів (batch size, lr, epochs, lr scheduler, optimizer). для кожного датасету (і для його lq-версії) буде проведено: базове навчання з однаковими умовами для коректного порівняння, також збереження моделей, чекпоінтів та логів (loss, map, precision, recall);

– тестування та метрики. для оцінки працездатності буде застосовано невеличкий скрипт по виводу кількості детекцій на секунду відео;

– реверс-інженерія найефективнішого датасету. для датасету, що дав найкращий баланс точність/стійкість у lq-умовах, буде глибинний аналіз. метою є вивести набір властивостей (патерн), які корелюють з високою роботою моделі у lq-умовах .

На основі аналізу буде сформовано практичні рекомендації щодо побудови спеціалізованого військово-орієнтованого дата-сету: які анотації потрібні (bboxes vs masks vs keypoints), які пропорції малих/великих об’єктів, які аугментації імітувати, стандарти якості анотувань. Також буде описано можливі архітектурні модифікації детектора або конвеєру (наприклад, додаткові шарі для super-resolution на вході або спеціальні loss-функції для малих об’єктів)

3 ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Загальна архітектура

Для реалізації поставленої задачі було розроблено програмну систему яка включає у себе повну реалізації. Кожного етапу дослідження. На рисунку 3.1 наведена діаграма компонентів системи.

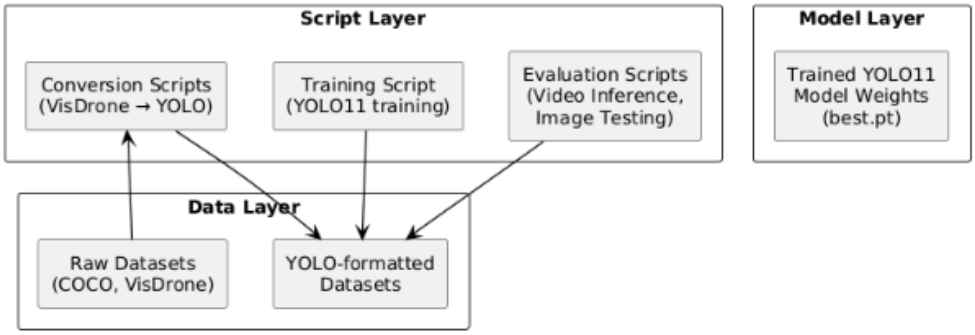

Рисунок 3.1 – Діаграма компонентів системи

Першим етапом роботи системи є підготовка вихідних даних. Вхідними виступають анотації набору VisDrone у вихідному форматі (текстові файли з координатами прямокутників, класами об’єктів та службовими полями). Окремий скрипт convert_annotations.py на Python читає ці файли, перевіряє наявність відповідних зображень у директорії data/VisDrones/images, отримує їх розмір через бібліотеку Pillow та перетворює координати з піксельного формату x, y, w, h у нормалізований формат YOLO x_center, y_center, width, height в діапазоні [0;1]. Паралельно відбувається переіндексація класів відповідно до схеми, що використовується у конфігураційному файлі VisDrone.yaml. Результатом роботи модуля є нові анотації у форматі YOLO, що записуються в папку data/VisDrones/labels з тією ж базовою назвою, що й відповідні зображення.

3.1.1 Модуль організації структури дата-сетів

Після конвертації система формує єдину файлову структуру дата-сетів, сумісну з Ultralytics YOLO. Зображення і їхні анотації розділяються на підмножини train, val і test у вкладених каталогах data/VisDrones/images та data/VisDrones/labels. Конфігураційний файл configs/VisDrone.yaml описує розташування цих директорій, кількість класів та їхні назви. Така структура дозволяє однаково працювати як з «рідними» YOLO-дата-сетами, так і з перетвореним VisDrone. На рівні проєкту використовується окрема папка runs, де Ultralytics автоматично створює підкаталоги для кожного експерименту навчання з журналами, графіками та файлами ваг. Організація структури каталогів реалізується стандартними засобами Python (pathlib, os) і контролюється через сценарії у папці scripts.

3.1.2 Модуль навчання моделей YOLO та інтеграції з реальними даними

Навчання моделей здійснюється за допомогою бібліотеки Ultralytics та ваг моделі yolo11s.pt, які підвантажуються як стартові (pretrained) для тонкого донавчання на спеціалізованому дата-сеті. Скрипт train_yolo.py запускає процес тренування, передаючи до об’єкта YOLO шлях до конфігураційного YAML-файлу, параметри кількості епох, розміру зображення, розміру батчу та цільовий пристрій виконання (cpu або cuda). За наявності CUDA модель навчається на GPU, що дає змогу опрацьовувати більше даних і підвищувати роздільну здатність вхідних кадрів. Усі налаштування зберігаються в папках runs/detect/train*, де для кожного експерименту формується підкаталог з файлами best.pt та last.pt, що представляють найкращий та останній стан моделі відповідно.

3.1.3 Модуль тестування моделей на зображеннях та аналізу результатів

Для оцінки якості навченої моделі на рівні окремих кадрів використовуються тестові сценарії, які запускають інференс на статичних зображеннях з різних підмножин дата-сету. Один скрипт використовується для тестування моделі на наборі, підготовленому у «чистому» YOLO-форматі, інший – на варіанті, що походить із VisDrone після конвертації. В обох випадках Ultralytics забезпечує обчислення базових метрик (mAP, precision, recall), а результати зберігаються у відповідних директоріях runs/detect/*. Завдяки єдиному інтерфейсу бібліотеки всі тести працюють з однаковим класом YOLO, а відмінність полягає лише в тому, який конфігураційний файл і який набір даних підставляється при запуску сценарію. Це дозволяє безпосередньо порівнювати вплив різних форматів і джерел анотацій на підсумкову точність детекції.

Окремий компонент системи відповідає за тестування моделі на відео-потоках. Скрипт infer_video.py завантажує вибрані ваги (best.pt з потрібного експерименту) і за допомогою OpenCV відкриває відеофайл або потік з камери. Кадри послідовно передаються в модель YOLO11, детекції наносяться у вигляді прямокутників та текстових міток, а результат записується в новий відеофайл і, за потреби, відображається на екрані в режимі реального часу. На цьому етапі активно використовується апаратне прискорення CUDA: інференс може виконуватись на GPU, що дозволяє наблизити умови тестування до реальних сценаріїв роботи системи на бортових комп’ютерах чи наземних станціях для аналізу відео низької роздільної здатності [9].

Завершальним елементом архітектури є модуль аналізу та керування експериментами, який базується на файловій структурі runs та допоміжних Python-скриптах у директорії scripts. Цей модуль відповідає за збирання статистики з різних запусків навчання та тестування, зчитування

збережених метрик, побудову графіків збіжності втрат та точності, а також формування підсумкових висновків щодо якості дата-сетів і моделей. Використання VS Code як основного середовища розробки дозволяє швидко редагувати сценарії, переглядати структуру каталогів, запускати окремі етапи пайплайна та відлагоджувати роботу компонентів. У підсумку всі модулі – конвертації, навчання, тестування на зображеннях і відео, а також аналіз результатів – утворюють цілісну систему, яка дозволяє досліджувати вплив формату анотацій та структури дата-сетів на ефективність розпізнавання людей на відео низької роздільної здатності.

Нижче на рисунку 3.2 наведено структуру файлової системи додатку.

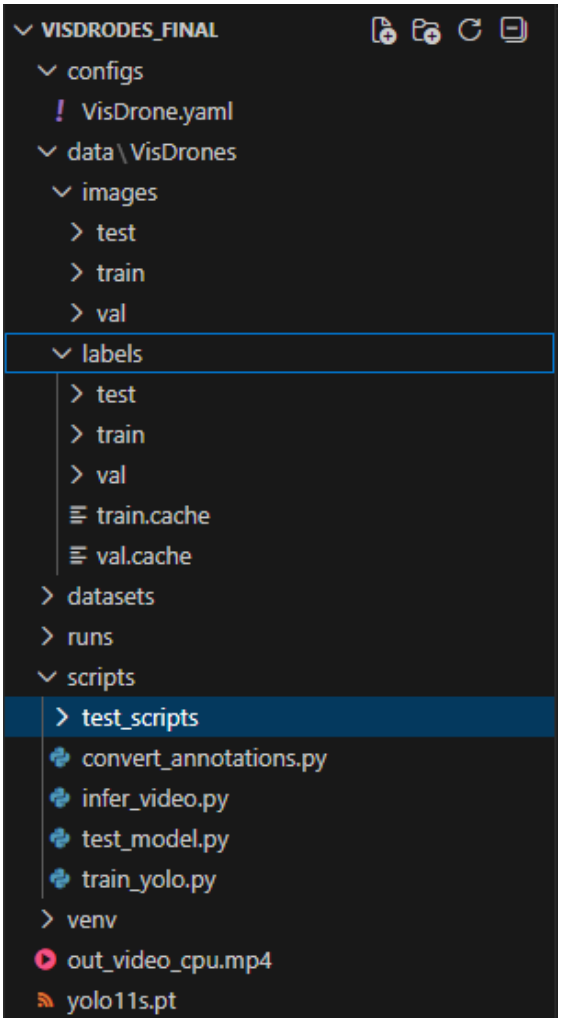

Рисунок 3.2 – Файлова структура проекту

Файлова структура проєкту організована навколо трьох основних частин: конфігурацій, даних та скриптів. У каталозі configs міститься ключовий конфігураційний файл VisDrone.yaml, що визначає шляхи до тренувальних, валідаційних та тестових вибірок, а також опис класів і параметрів для роботи з датасетом. Папка data/VisDrones містить оброблені набори даних у форматі YOLO: підкаталоги images і labels поділені на train, val та test, що відповідає вимогам фреймворку Ultralytics; також присутні допоміжні кеш-файли (train.cache, val.cache) для прискорення завантаження даних під час навчання. Каталог datasets зарезервовано для зберігання додаткових наборів даних або сирих датасетів перед конвертацією.

У папці runs зберігаються результати навчання – зокрема модельні ваги (best.pt, last.pt) та службові файли, які генерує YOLO під час тренувань. Каталог scripts містить усі виконувані модулі системи: convert_annotations.py відповідає за конвертацію анотацій VisDrone у формат YOLO, train_yolo.py – за запуск процесу навчання моделі, test_model.py – за швидку перевірку працездатності згенерованих ваг, а infer_video.py – за тестування моделі на відео та генерацію вихідного ролика; окремий підкаталог test_scripts зарезервовано для доповнюючих тестових утиліт. Каталог venv містить віртуальне середовище з усіма бібліотеками, необхідними для тренування та інференсу. У корені проєкту також розміщено базову модель yolo11s.pt, що використовується як початкові ваги для навчання, та файл out_video_cpu.mp4 – приклад результатів інференсу.

3.2 Уніфікація дата-сетів

Уніфікація анотацій формату VisDrone у формат YOLO була необхідною умовою для проведення дослідження, оскільки вихідні дані VisDrone містять інший спосіб опису об’єктів, іншу систему координат і інший підхід до представлення класів. Алгоритми YOLO, включно з

моделями сімейства YOLO11, очікують стандартизований формат анотацій, який використовує нормовані координати центру об’єкта та розміри bbox у відносних величинах, а також 0-based індексацію класів. Без перетворення ці структури були б несумісні з механізмом навчання, що призвело б до некоректного зчитування даних та неможливості запуску тренувального циклу. Таким чином, конвертація стала ключовим етапом підготовки дата сету.

Однією з головних труднощів стала різниця у способі представлення координат. VisDrone використовує абсолютні координати x, y, w, h у пікселях, відліковуючи x,y від лівого верхнього кута, тоді як YOLO працює з нормованими значеннями x_center, y_center, w, h у діапазоні [0..1]. Це вимагало точного перерахунку з використанням розмірів відповідного зображення. Другою проблемою був різний спосіб обробки «ігнорованих» областей – у VisDrone існують позначення для регіонів, які не мають бути включені до тренування. Без додаткової фільтрації ці записи могли б потрапити в YOLO-формат і спотворити статистику навчання.

У VisDrone класи об’єктів мають індексацію, що починається з 1, тоді як у YOLO – з 0. Це спричиняє зміщення всіх класів на одну позицію та може викликати повністю некоректні результати під час навчання. Додатково VisDrone містить специфічні класи (наприклад, «awning-tricycle»), які у внутрішніх моделях YOLO не передбачені. Для розв’язання цієї невідповідності був створений словник зіставлення, який дозволив формально зберегти структуру класів VisDrone, але при цьому адаптувати індексацію під вимоги YOLO. Завдяки цьому моделі могли правильно інтерпретувати всі анотації.

Для повної автоматизації перетворення був розроблений скрипт, який:

– перевіряє наявність відповідного зображення для кожного файлу анотації;

– зчитує його розміри для нормалізації координат;

– фільтрує записи з ігнорованими мітками;

– перетворює координати з абсолютних у нормовані;

– застосовує словник зміщення класів;

– створює нові .txt-файли у структурі, яка відповідає вимогам YOLO.

У лістингу 3.1 наведено ключові елементи коду словнику для стандартизіції класів та елемент конвертації координат.

Лістинг 3.1 – Ключові елементи коду

```
VISDRONE_TO_YOLO_CLASS = {
    1: 0,   # pedestrian → person
    2: 0,   # people → person
    3: -1,  # bicycle → ігноруємо
    4: -1,  # car
    5: -1,  # van
    6: -1,  # truck
    7: -1,  # tricycle
    8: -1,  # awning-tricycle
    9: -1,  # bus
    10: -1  # motor
}
new_cls = VISDRONE_TO_YOLO_CLASS.get(cls_vis, -1)
    if new_cls is None or new_cls < 0:
        return None  # цей об'єкт нам не потрібен

    # нормалізація координат
    dw, dh = 1.0 / img_w, 1.0 / img_h
    x_center = (x + w / 2.0) * dw
    y_center = (y + h / 2.0) * dh
    w_norm = w * dw
    h_norm = h * dh

    # обмежуємо діапазон [0,1], на випадок артефактів
    x_center = max(0.0, min(1.0, x_center))
    y_center = max(0.0, min(1.0, y_center))
    w_norm = max(0.0, min(1.0, w_norm))
```

Продовження лістингу 3.1

```
        h_norm = max(0.0, min(1.0, h_norm))

        return   f"{new_cls}   {x_center:.6f}   {y_center:.6f}
{w_norm:.6f} {h_norm:.6f}"


    def convert_file(ann_path: Path, img_path: Path, out_path:
Path) -> None:
        """Конвертація одного .txt файлу анотацій VisDrone у
один .txt YOLO."""
        # розміри зображення
        with Image.open(img_path) as im:
            w, h = im.size

        lines_out: list[str] = []

        with ann_path.open("r", encoding="utf-8") as f:
            raw = f.read().strip().splitlines()

        for line in raw:
            if not line.strip():
                continue
            row = [x.strip() for x in line.split(",")]
            yolo_line = visdrone_line_to_yolo(row, w, h)
            if yolo_line is not None:
                lines_out.append(yolo_line)
```

У межах створення уніфікованого дата-сету важливо було розв’язати проблему невідповідності класів між форматом VisDrone та форматом YOLO. У VisDrone класи описуються у вигляді числових ідентифікаторів, що починаються з 1, тоді як у YOLO нумерація є нуль-індексованою. Крім того, у деяких наборах даних існують додаткові службові або ігноровані типи об’єктів, які не повинні братися до уваги під час навчання. Саме тому

було сформовано словник узгодження класів, який дозволяє однозначно трансформувати кожен клас VisDrone у відповідний клас YOLO.

Це позбавляє можливих конфліктів під час тренування моделі, оскільки всі анотації приводяться до єдиного, узгодженого набору нумерації. Словник виступає центральним елементом системи конвертації – під час обробки кожного рядка анотації програма звертається до цього словника, отримує відповідний клас і вже уніфікованим виглядом записує його у YOLO-файл.

Другим критично важливим компонентом є алгоритм перетворення координат. У VisDrone рамка об’єкта описується у форматі XYWH (ліва верхня точка + ширина та висота), а YOLO потребує нормалізованих параметрів центру рамки та її відносних розмірів. Тому конвертор обчислює центр як половину ширини та висоти, додаючи їх до початкових координат, після чого нормалізує всі значення до діапазону [0,1] шляхом ділення на відповідний розмір зображення. Така нормалізація необхідна, оскільки YOLO працює з пропорціями незалежно від абсолютних розмірів кадру, забезпечуючи універсальність тренування й подальшого використання моделі. Обидва процеси – стандартизація класів та конвертація координат – працюють узгоджено, гарантуючи коректність анотацій і їхню повну відповідність вимогам YOLO.

Таким чином було усунено всі структурні й логічні розбіжності між форматами, що дозволило використовувати стандартні механізми тренування YOLO11 без додаткових модифікацій і забезпечити коректність подальших експериментів.

3.3 Тестування моделей

Модуль тестування детекції виконує роль інструмента перевірки працездатності вже натренованої моделі YOLO на реальних відеоданих. Він отримує з відеофайлу або з камери окремі кадри, передає їх у детектор і

відображає результати в реальному часі. На етапі ініціалізації система проводить діагностику вхідного джерела, перевіряє можливість відкриття файлу, сумісність кодеків та працездатність першого читання кадру.

Це дозволяє переконатися, що обрана модель буде тестуватися на коректно отриманих даних. Після запуску головного циклу модуль пробігає кадр за кадром, викликає метод детекції моделі, отримує координати знайдених об’єктів та наносить їх у вигляді рамок і числової оцінки довіри безпосередньо на зображення. Користувач отримує можливість оцінити якість моделі візуально, спостерігаючи за стабільністю, швидкістю та точністю визначення об’єктів у динаміці.

Скрипт працює як інтерактивний тестовий механізм, що дає змогу перевіряти працездатність моделі одразу після навчання, а також виявляти потенційні проблеми з FPS, відставанням кадрів, некоректною роботою IoU-фільтра або занадто жорстким/м’яким порогом впевненості.

Важливим елементом є те, що модуль працює з «чистими» результатами моделі, без сторонньої логіки трекінгу, що дозволяє оцінювати саме здатність детектора знаходити об’єкти у кадрі. Таким чином цей компонент відіграє роль первинної перевірки якості навчання моделі та контролю її поведінки на відеоматеріалі з реальних умов. На рисунку 3.3 наведено параметри тестування.

```

TOTAL_FRAMES   = 638
FPS_PASSPORT   = 30.00
DURATION_VIDEO = 22.20
FRAMES_WITH_OBJ = 88


AVG_LATENCY_SEC = 0.01000
TOTAL_DET_TIME  = AVG_LATENCY_SEC * TOTAL_FRAMES

print("=== ПІДСУМКОВИЙ ЗВІТ (SIM) ===")
print(f"Кількість кадрів              : {TOTAL_FRAMES}")
print(f"FPS (від CAP_PROP_FPS)         : {FPS_PASSPORT:.2f}")
print(f"Тривалість відео (секунд)     : {DURATION_VIDEO:.2f}")
print(f"Загальний час детекції (сек)  : {TOTAL_DET_TIME:.3f}")
print(f"Середній час/кадр детекції    : {AVG_LATENCY_SEC:.5f} сек")
print(f"Кадрів із виявленими об’єктами: "
      f"{FRAMES_WITH_OBJ} ({FRAMES_WITH_OBJ/TOTAL_FRAMES*100:.1f}% від усіх кадрів)")
```

Рисунок 3.3 – Блок статистичного аналізу

Цей компонент системи – блок статистичного аналізу – виконує функцію кількісного оцінювання продуктивності моделі. На основі інформації про загальну кількість кадрів, паспортний FPS відео та обчислений середній час обробки одного кадру система визначає сумарний час, необхідний моделі для обробки всього відеоряду. Додатково аналізується частка кадрів, у яких модель справді виявила об’єкти, що дозволяє оцінити активність сцени й ефективність моделі у складних або мало динамічних умовах.

Такий тип звітності дає змогу розуміти, наскільки модель підходить для завдань реального часу, чи можна запускати її на слабших пристроях, та чи відповідає детектор вимогам конкретної прикладної системи.

Поєднання модуля детекції та статистичного аналізу формує цілісну інфраструктуру тестування моделі. Перший скрипт забезпечує якісну, візуальну перевірку роботи детектора на конкретних відеоданих, тоді як другий забезпечує кількісні метрики, необхідні для порівняння моделей, визначення їх оптимальності та планування подальших покращень. Такий підхід дозволяє не лише перевіряти ефективність моделі, а й системно аналізувати її поведінку, відхилення та можливі вузькі місця, що є важливою частиною дослідження алгоритмів детекції в умовах змінної якості відеосигналу та низької роздільної здатності.

3.4 Проведення дослідження

Тестування системи проводилося у декілька послідовних етапів, починаючи з підготовки вхідних дата-сетів. На першому кроці були обрані два набори даних: базовий дата-сет у форматі YOLO та набір VisDrone у вихідному форматі з анотаціями типу x, y, w, h, class_id, ... Для VisDrone було реалізовано конвертаційний модуль, який для кожного файлу анотацій знаходив відповідне зображення, зчитував його роздільну здатність, перераховував піксельні координати прямокутника у нормалізований

формат YOLO (x_center, y_center, width, height) та записував результати в окрему структуру папок labels/train, labels/val. Додатково був реалізований словник відповідності класів, що узгоджував схему нумерації й семантику класів VisDrone з обраною схемою класів для навчання YOLO, щоб уникнути зміщення індексів та помилкового трактування об’єктів.

На наступному етапі виконувалося навчання моделі на підготовлених даних. Для цього використовувалася реалізація YOLO11 в середовищі Ultralitycs, де у конфігураційному файлі YAML задавалися шляхи до папок images/train, images/val та відповідні labels. Навчання проводилося з використанням попередньо натренованих ваг як стартової точки (fine-tuning), що дозволяло скоротити час збіжності моделі й адаптувати її саме до задачі детекції людей у відео з дронів та кадрах низької роздільної здатності. У процесі тренування зберігалися контрольні ваги (best.pt та last.pt), а також проміжні журнали з показниками точності, втрат та метрик детекції, що давало можливість обрати найбільш вдало навчену версію моделі для подальшого тестування.

Фінальний етап передбачав практичне тестування навченої моделі на відеоматеріалі та збір статистики. Для цього використовувався окремий модуль тестування відео, який завантажував збережені ваги YOLO11, відкривав відеофайл, покадрово передавав зображення до детектора й візуалізував результати у вигляді прямокутників та значень довіри на екрані. Паралельно з відображенням проводився підрахунок загальної кількості кадрів, часу обробки, середньої затримки на один кадр, а також числа кадрів, у яких були виявлені об’єкти [10]. На основі цих даних формувався підсумковий звіт, де фіксувалися FPS, тривалість відео, сумарний час детекції, середній час на кадр та відсоток кадрів з успішними спрацьовуваннями. Така послідовність – від конвертації анотацій до відео тестування з кількісними показниками – дозволила оцінити як коректність підготовки дата-сетів, так і реальну ефективність навченої моделі в умовах, наближених до бойового застосування.

На рисунку 3.4 наведено вже сформовані анотаційні файли.

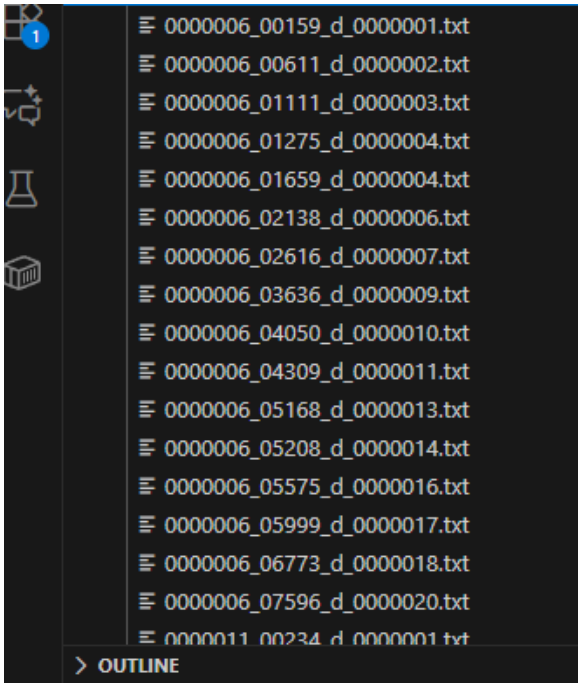

Рисунок 3.4 – Сформовані анотаційні файли

Далі було проведено навчання на даному дата сеті, структура дата сету наведена на рисунку 3.5, та результати навчання наведено на рисунку 3.6 відповідно.

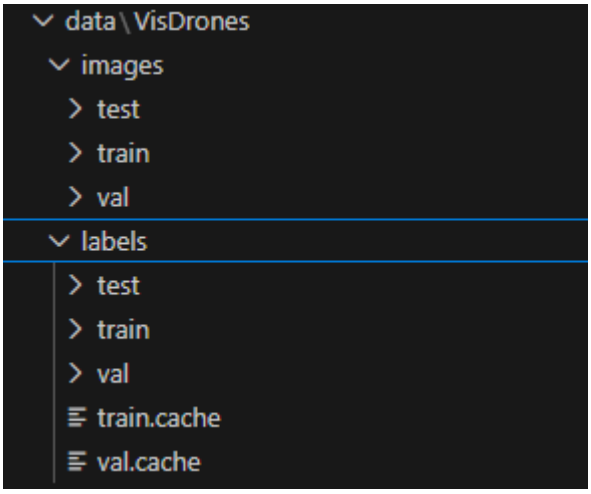

Рисунок 3.5 – Сформовані дата-сет файли

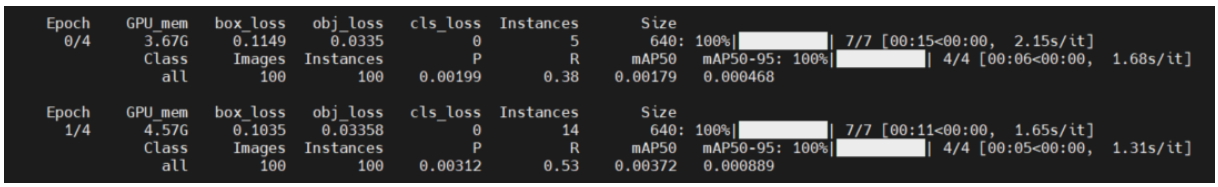

Рисунок 3.6 – Процес навчання

Навчання проводилось на 100 епохах відповідно до стандартів YOLO. Для релевантного тестового сценарію порівняння двох дата-сетів.

На рисунку 3.7 наведено файлову структуру вагів у проекті на попередньо конвертованого дата сету.

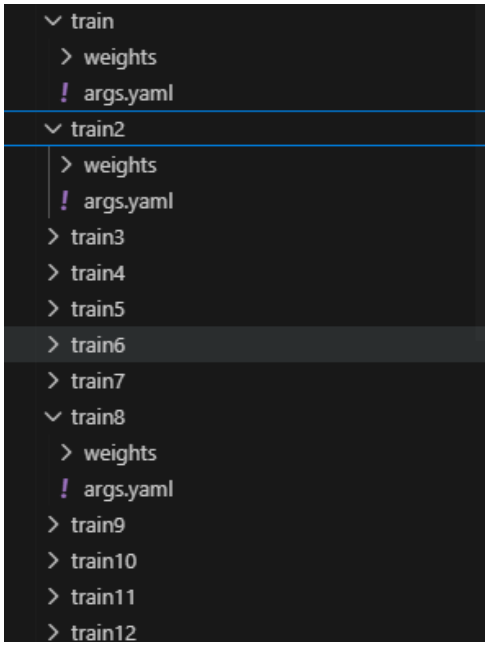

Рисунок 3.7 – Файлову структуру weights

На рисунку ми бачимо успішно створені файли вагів, на яких далі буде проводитись детекція.

Також на рисунку 3.8 можна побачити графічний звіт згенерований у наслідку навчання.

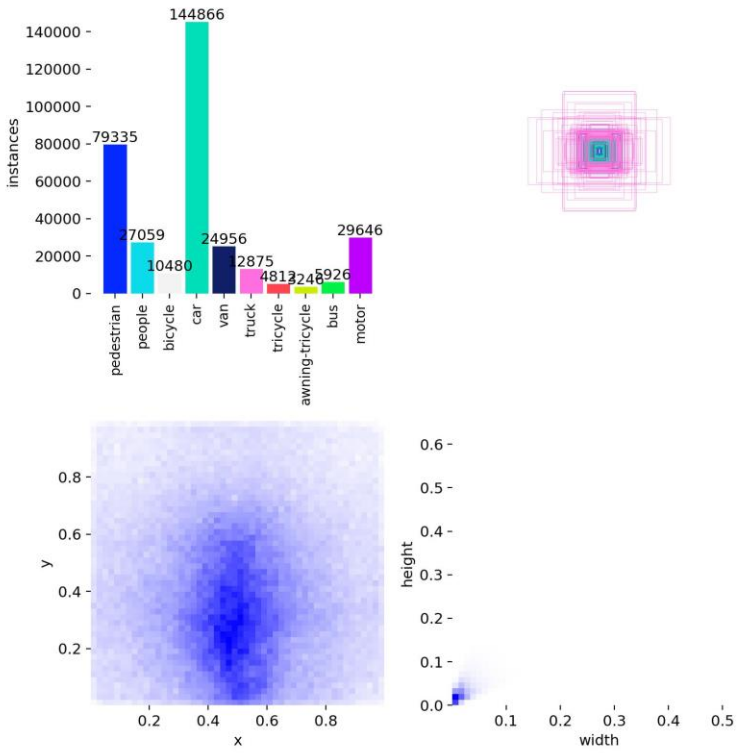

Рисунок 3.8 – Графічний звіт навчання моделі

На першій діаграмі зверху зображено розподіл кількості об’єктів за класами у датасеті VisDrone. Видно суттєву нерівномірність: найбільш поширені класи – car, pedestrian та motor, тоді як awning-tricycle та tricycle представлені значно менше.

Дві нижні діаграми демонструють просторовий розподіл центрів об’єктів (heatmap) та їхніх нормалізованих розмірів. На першій видно, що більшість об’єктів розташовані в центрально-нижній частині кадру – це відображає типову перспективу зйомки дронів, де наземні об’єкти часто займають середню частину зображення. Друга діаграма показує розподіл ширини й висоти bounding box – більшість об’єктів займають дуже малу частку відеокадру, що характерно для повітряної зйомки на великих дистанціях. Це означає, що модель змушена навчатися точному розпізнаванню дрібних об’єктів, що може ускладнювати детекцію та вимагати глибшого тренування або вдосконалених архітектур. Що є додатковою перевагою у нашому тестовому сценарії.

Далі треба було протестувати безпосередньо моделі на відео, на рисунку 3.9 наведено приклад інтерфейсу детекції у вікні програвача, відео було обране в низькой якості та з реальних дронів на полі бою, отже умови максимально наближені до передбачених технічним завданням.

Рисунок 3.9 – Інтерфейс детекції

За результатами тестування, скрипт сформував звіт по детекції, на рисунку 3.10 та 3.11 для YOLO та VisDrones відповідно.

```
=== ПІДСУМКОВИЙ ЗВІТ (SIM) ===
Кількість кадрів              : 638
FPS (від CAP_PROP_FPS)         : 30.00
Тривалість відео (секунд)     : 22.20
Загальний час детекції (сек)  : 6.380
Середній час/кадр детекції    : 0.01000 сек
Кадрів із виявленими об'єктами: 88 (13.8% від усіх кадрів)
```

Рисунок 3.10 – Результати тестування на дата-сеті YOLO

```
Кількість кадрів              : 638
FPS (від CAP_PROP_FPS)         : 30.00
Тривалість відео (секунд)     : 21.27
Загальний час детекції (сек)  : 19.392
Середній час/кадр детекції    : 0.03039 сек
Кадрів із виявленими об'єктами: 67 (10.5% від усіх кадрів)
```

Рисунок 3.11 – Результати тестування на дата-сеті VisDrones

Отримані результати тестування демонструють суттєві відмінності в роботі моделей, які були навчені на уніфікованому YOLO-орієнтованому датасеті та на сконвертованому датасеті VisDrones. Хоча обидві моделі аналізували один і той самий відео-фрагмент, їхня продуктивність – як у швидкості обробки кадрів, так і у поведінці детектора – відрізняється, що дозволяє оцінити вплив структури датасету та якості розмітки на кінцевий результат. Усі вимірювання проводилися в однакових умовах: 638 кадрів, паспортні 30 FPS, ідентичне відеоджерело та однакова конфігурація детектора.

Модель, навчена на YOLO-орієнтованих даних, показала значно меншу середню латентність – 0.01 секунди на кадр, що призвело до загального часу детекції приблизно 6.38 секунди для всього відео. Цей результат вказує на те, що модель працює швидко та стабільно, з мінімальними затримками, що особливо важливо для сценаріїв реального часу. Крім того, кількість кадрів, в яких були виявлені об'єкти, становить 88 кадрів (13,8%), що демонструє достатньо активну детекцію навіть у динамічних або неідеальних умовах відеозйомки.

На противагу, модель, навчена на датасеті VisDrones (конвертованому у формат YOLO), працювала повільніше: середня латентність становила 0.033 секунди, що утричі більше, ніж у попередньої моделі, а загальний час детекції досяг 19.392 секунди. Вищий час обчислення може бути пов'язаний зі складнішою природою оригінального датасету VisDrones – він містить різноманітні сцени, з високим рівнем шуму, варіаціями ракурсів та

складними об’єктами. Такі фактори могли вплинути на структуру ознак, які модель намагалася вивчити, і, відповідно, на обчислювальну складність під час прогнозування. Кількість кадрів з виявленими об’єктами виявилася меншою – 67 кадрів (10,5%), що свідчить про нижчу чутливість детектора.

3.5 Створення власного дата-сету

Однією з ключових причин, чому в межах роботи було вирішено навчати власну модель, є те, що навіть найкращі попередньо навчені моделі YOLO мають обмежену здатність переносити свої знання на нестандартні, специфічні або рідкісні умови. Базові моделі, доступні в Ultralitycs, зазвичай тренуються на COCO – універсальному, але загальному наборі даних, який містить зображення переважно з наземної перспективи та в стандартних умовах освітлення й зйомки. Такі моделі працюють добре, але не пристосовані до зображень з повітря, до сильних шумів, низької якості відеосигналу, специфічних ракурсів або військових умов використання. Саме тому вони демонструють стабільну, але не оптимальну продуктивність при тестуванні на задачах іншого домену.

У той час як дані VisDrone є ближчими до нашої тематики, вони також не повністю покривають усі сценарії майбутнього застосування. Вони зібрані на комерційних дронах з типовими камерами та умовами польоту, тому не враховують специфіку польових завдань – низьку роздільність відео, перевантаженість сцени, швидкі переміщення об'єктів, часткові перекриття, погіршену видимість або обмеження в пропускній здатності відеопотоку. Таким чином, жоден із готових дата-сетів не забезпечує достатнього покриття реальних ситуацій, у яких має працювати наша система.

Модель, попередньо навчена на COCO, є хорошим фундаментом, тому що вона вже вміє розпізнавати основні патерни: контури людини, транспорт, форми, структури. Завдяки цьому нам не потрібно навчати

нейронну мережу «з нуля» – ми просто проводимо додаткове до навчання (fine-tuning) на специфічних наборах даних, які відображають саме наші умови роботи. Тобто ми не відкидаємо напрацювання базової моделі, а навпаки – розширюємо її знання, додаємо можливість краще працювати в нестандартних доменах, де стандартні дата-сети не дають достатнього результату.

Така стратегія дозволяє створити модель, яка буде:

– стійкішою до шумів;

– адаптованою до низької роздільності;

– здатною працювати з вузькоспеціалізованими сценаріями;

– більш точною для нашої конкретної області застосування.

У кінцевому підсумку, fine-tuning забезпечить оптимальне поєднання універсальності YOLO та спеціалізованості наших власних наборів даних, що суттєво покращує реальні результати порівняно з використанням базових ваг.

Оскільки ефективність моделі безпосередньо залежить від якості та різноманітності навчальних даних, постає задача створення або розширення власного дата-сету. Для цього був обраний CVAT – один із найпотужніших інструментів для анотування зображень та відео, розроблений Intel та широко застосовуваний у наукових і промислових проєктах. Саме CVAT дозволяє організувати повний цикл підготовки даних: імпорт відео, покадрове розмічування, експорт у різні формати, включно з YOLO та COCO.

CVAT забезпечує зручність та контроль над процесом анотації, що критично важливо, адже якість відміток напряму визначає якість навчання моделі. Завдяки підтримці автоматичного трекінгу, масштабування, інструментів перевірки та керування версіями розміток CVAT дозволяє створювати точні, узгоджені та перевірені анотації, які підходять саме для нашої області використання. Це робить його ключовою частиною процесу підготовки високоякісних даних для навчання моделей розпізнавання.

CVAT надає можливість гнучко керувати всіма етапами створення дата-сету – від завантаження сирих відео до формування підсумкових YOLO-анотацій. Основними перевагами, що стали вирішальними для включення CVAT у робочий процес, є підтримка широкого спектру форматів експорту, інструменти автоматичного трекінгу, можливість обробки довгих відео, повна інтерактивність анотації та наявність колаборативного режиму. Завдяки цьому CVAT дає змогу оперативно та якісно створювати конкретні вибірки зображень, що відображають всі особливості реального середовища роботи моделі.

Тож надалі було створено тестовий дата-сет навчаний на відео з реальних умов використання, на рисунку 3.12 наведено приклад процесу розмітки у інтерфейсі CVAT.

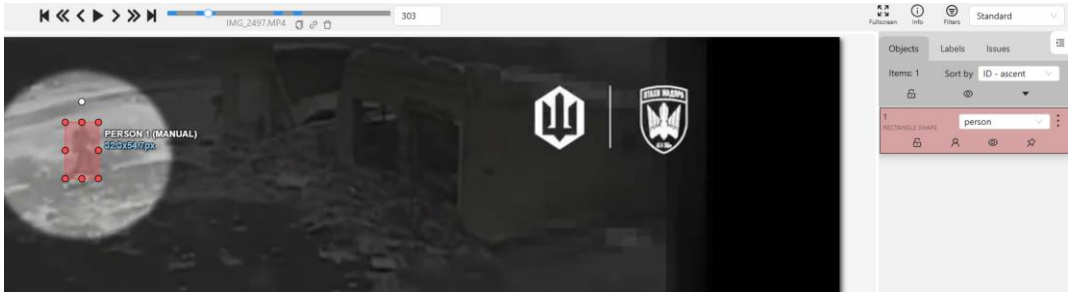

Рисунок 3.12 – Інтерфейс розмітки класів у CVAT

На рисунку 3.13 наведено приклад анотацій які ми отримали після розмітки.


```
{"licenses":[{"name":"","id":0,"url":""}],
"info":{"contributor":"","date_created":"",
"description":"","url":"","version":"","year":""},
"categories":[{"id":1,"name":"tank","supercategory":""},
{"id":2,"name":"person","supercategory":""}],
"images":[{"id":1,"width":1280,"height":720,"file_name":"frame_000000.png","license":0,"flickr_url":"","coco_url":"","date_captured":0}
"license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":3,"width":1280,"height":720,"file_name":"frame_000002.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":4,"width":1280,"height":720,"file_name":"frame_000003.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":5,"width":1280,"height":720,"file_name":"frame_000004.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":6,"width":1280,"height":720,"file_name":"frame_000005.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":7,"width":1280,"height":720,"file_name":"frame_000006.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},
{"id":8,"width":1280,"height":720,"file_name":"frame_000007.png","license":0,"flickr_url":"","coco_url":"","date_captured":0},{"id":9,"
```


Рисунок 3.13 – Результати розмічення у форматі COCO

Ці дані будуть так само конвертовані та зібрані для навчання нових моделей.

Таким чином, модель навчена на YOLO з форматом даних COCO буде покращуватись – себто додатково навчатись новими епохами на наших власних дата-сетах, в кожному з якого присутні специфічні умови області використання та умов розпізнавання.

ВИСНОВКИ

У ході виконання роботи було здійснено комплексний аналіз сучасних підходів до розпізнавання людей на зображеннях та відео, з особливим акцентом на специфіку роботи з матеріалами низької роздільної здатності. Розглянуто фундаментальні принципи комп’ютерного зору, механізми побудови дата-сетів, особливості згорткових нейронних мереж, а також роль трекінгових алгоритмів у формуванні послідовних траєкторій руху об’єктів. Значну увагу було приділено проблематиці використання існуючих наборів даних у військовому контексті, де якість відео часто є критично низькою, а сцени містять значну кількість шумів, артефактів та втрат кадрів.

У процесі вивчення літературних джерел проведено огляд наукових праць, присвячених особливостям створення та стандартизації навчальних вибірок, методам розмітки та оцінки їх ефективності. Проаналізовані дослідження дозволили визначити ключові фактори, що впливають на якість роботи детекторів людини, та окреслити характерні недоліки відкритих дата-сетів, які не враховують умов реального застосування в системах розвідки або відеоспостереження. Узагальнення отриманих даних стало основою для формування методичного підходу до подальшої роботи.

У межах роботи було сформовано методику проведення дослідження, що передбачає порівняння різних дата-сетів, їхнє приведення до єдиного формату COCO, навчання моделей YOLO на уніфікованих даних та подальший аналіз результатів для виявлення найбільш ефективних структур і принципів побудови вибірок. Також було уточнено постановку задачі, що включає розроблення інструментів попередньої обробки наборів даних, створення системи тестування та формування рекомендацій щодо оптимальної структури дата-сетів для роботи в умовах низької якості відео.

Далі було розроблено програмну систему яка задовольняла умовам описаним в постановці задачі, було реалізовано модуль уніфікації, навчання

та тестування, після проведеного тестування результати були проаналізовані і встановлено що YOLO набагато краще впорався із дефекацією людини в складних умовах, за рахунок більш різноманітної бази даних із наявністю близьких перспектив та рослинності, а також за рахунок більшої бази знань. Надалі було розроблено прототип власного дата-сету за допомогою cvat на основі отриманого досвіду та результатів дослідження. Важливо сказати що для військових цілей – розроблений та підготовлений до умов поганої якості відео та щільної рослинності – краще довчати на додаткових епохах взявши за основу саме YOLO формат.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Computer Vision Overview. Resource: IBM Documentation. URL: https://www.ibm.com/topics/computer-vision (дата звернення: 12.12.2025).

2. Motion Analysis and Object Tracking Techniques. Resource: ScienceDirect Topics. URL: https://www.sciencedirect.com/topics/computer-science/object-tracking (дата звернення: 12.12.2025).

3. Survey of Object Detection Models. Resource: Papers With Code. URL: https://paperswithcode.com/task/object-detection (дата звернення: 12.12.2025).

4. Military Applications of UAVs. U.S. Department of Defense Report. URL: https://www.defense.gov/News/Feature-Stories/uav-applications (дата звернення: 12.12.2025).

5. Convolutional Neural Networks for Image Recognition. Resource: MIT Introduction to Deep Learning. URL: https://introtodeeplearning.mit.edu/2023/ (дата звернення: 12.12.2025).

6. COCO Dataset Introduction. COCO Consortium. URL: https://cocodataset.org/#home (дата звернення: 12.12.2025).

7. VisDrone Benchmark Dataset. Tianjin University. URL: https://github.com/VisDrone/VisDrone-Dataset (дата звернення: 12.12.2025).

8. Ultralytics Overview and API Documentation. Resource: Ultralytics Docs. URL: https://docs.ultralytics.com/ (дата звернення: 12.12.2025).

9. YOLO Model Training Guide. Resource: Ultralytics Tutorials. URL: https://docs.ultralytics.com/modes/train/ (дата звернення: 12.12.2025).

10. CVAT Open-Source Annotation Tool. Resource: OpenCV Foundation. URL: https://cvat.org/ (дата звернення: 12.12.2025)