

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ) _____ інформаційних технологій _____

ПОГОДЖЕНО
Декан факультету (Директор ННІ)
інформаційних технологій

(назва факультету (ННІ))

Ігор БОЛБОТ

(підпис)

(ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 20_ р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних наук

(назва кафедри)

Белла ГОЛУБ

(підпис)

(ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 20_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Інтелектуальної системи розпізнавання об'єктів на основі комп'ютерного зору для автоматизації управління складськими запасами

Спеціальність _____ 122 "Комп'ютерні науки" _____

(код і найменування)

Освітня програма _____ Інформаційні управляючі системи та технології _____

(назва)

Орієнтація освітньої програми _____ освітньо-професійна _____

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

К.Т.Н., доцент

(науковий ступінь та вчене звання)

(підпис)

Белла ГОЛУБ

(ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

д.ек.н., проф.

(науковий ступінь та вчене звання)

(підпис)

Роман РУДЕНСЬКИЙ

(ім'я ПРІЗВИЩЕ)

Виконав

(підпис)

Дмитро ПЕТРИЧЕНКО

(ім'я ПРІЗВИЩЕ студента)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) _____ інформаційних технологій _____

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук
доцент, к.т.н. _____ Белла ГОЛУБ
(науковий ступінь, вчене звання) (підпис) (ПІБ)
"01" листопада 2024 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Петриченко Дмитро Володимирович.

(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітня програма Інформаційні управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Інтелектуальної системи розпізнавання об'єктів на основі комп'ютерного зору для автоматизації управління складськими запасами

затверджена наказом ректора НУБіП України від " 1 " листопада 2024р. №1964 «С»
Термін подання завершеної роботи на кафедру 11 листопада 2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: Нормативно-довідкова та наукова література з інформаційних систем, баз даних, методів візуалізації та обробки даних; технічна документація з проектування та розробки програмних комплексів; програмні засоби для розробки веб-застосунків і інтерактивних інтерфейсів; вимоги стандартів до побудови інформаційних систем і користувацьких інтерфейсів.

Перелік питань, що підлягають дослідженню:

1. Системний аналіз предметної області
2. Моделювання та архітектурне проектування системи
3. Реалізація програмного забезпечення та технологічна інфраструктура системи
4. Тестування та оцінювання ефективності системи

Перелік графічного матеріалу (за потреби) презентація, постер, схеми та діаграми архітектури системи

Дата видачі завдання " 1 " листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Роман РУДЕНСЬКИЙ

(підпис)

(ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Дмитро ПЕТРИЧЕНКО

(підпис)

(ім'я ПРІЗВИЩЕ студента)

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ..... | 6 |
| ВСТУП..... | 8 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ | 11 |
| 1.1 Опис предметної області | 11 |
| 1.2 Теоретико-методологічні засади та стан наукових досліджень..... | 14 |
| 1.3 Аналіз існуючих рішень | 19 |
| 1.4 Моделювання предметної області | 24 |
| 1.5 Аналіз вимог інформаційної системи | 28 |
| 1.6 Постановка завдання | 31 |
| 1.7 Висновки до першого розділу..... | 32 |
| 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 35 |
| 2.1 Логічна модель даних розпізнавання об'єктів складських підприємств | 35 |
| 2.2 Діаграма класів та кооперації інформаційної системи | 37 |
| 2.3 Діаграма компонентів інформаційної системи | 42 |
| 2.4 Діаграма пакетів | 45 |
| 2.5 Висновки до другого розділу | 46 |
| 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 49 |
| 3.1 Вибір технологій та інструментальних засобів реалізації системи . | 49 |

| | |
|--|----|
| 3.2 Архітектура системи, проектування функціоналу результатів дослідження | 51 |
| 3.3 Інформаційна база системи | 54 |
| 3.3 Висновки до третього розділу | 56 |
| 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ | 59 |
| 4.1 План тестування програмних модулів та методика оцінювання результатів | 59 |
| 4.2 Тестування інтелектуальної системи розпізнавання об'єктів у середовищі програмного емулятора | 61 |
| 4.3 Результати тестування та аналіз ефективності системи | 64 |
| 4.4 Висновки до четвертого розділу | 66 |
| ВИСНОВКИ | 68 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 70 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

1. ACCS – Access Control and Counting System, система контролю доступу та трекінгу відвідуваності.
2. API – Application Programming Interface, програмний інтерфейс взаємодії.
3. DB – Database, база даних.
4. ESP32 – Embedded System Platform 32-bit, мікроконтролер з Wi-Fi/BLE модулем.
5. HTTP(S) – HyperText Transfer Protocol (Secure), протокол передавання гіпертексту (захищений).
6. JSON – JavaScript Object Notation, формат представлення структурованих даних.
7. KPI – Key Performance Indicator, ключовий показник ефективності.
8. LAN – Local Area Network, локальна обчислювальна мережа.
9. MQTT – Message Queuing Telemetry Transport, телеметричний протокол обміну повідомленнями.
10. MQTT QoS1 – Quality of Service Level 1, гарантія доставки «принаймні один раз».
11. p95 / p99 – перцентилі затримок обробки (95-й і 99-й відповідно).
12. REST – Representational State Transfer, архітектурний стиль побудови веб-API.
13. RSSI – Received Signal Strength Indicator, рівень потужності прийнятого сигналу.
14. SQL – Structured Query Language, мова структурованих запитів.
15. TLS – Transport Layer Security, протокол захисту транспортного рівня.
16. UI – User Interface, інтерфейс користувача.
17. UID – User Identifier, унікальний ідентифікатор користувача.

18. UUID – Universally Unique Identifier, глобальний унікальний ідентифікатор.
19. WAN – Wide Area Network, глобальна мережа.
20. ЗЦД – зона контролю доступу (логічна область з політиками авторизації).
21. ПС – програмна система.
22. СУБД – система управління базами даних.
23. ТЗ – технічне завдання.
24. ТЛС-канал – захищений канал передавання даних за протоколом TLS.

ВСТУП

Інтенсивна цифровізація логістичних процесів та зростання вимог до оперативності оброблення товарних потоків актуалізують потребу у впровадженні інтелектуальних технологій автоматизації складських операцій. Традиційні системи управління запасами здебільшого ґрунтуються на ручному введенні даних або використанні додаткових маркерів, що потребує значних людських ресурсів, знижує продуктивність та призводить до накопичення помилок у звітності. У сучасних високонавантажених складах, де обробляються тисячі одиниць продукції, особливо важливими стають методи, здатні забезпечити безперервне розпізнавання товарів, контроль переміщень, моніторинг залишків та відстеження операцій у режимі реального часу. У цьому контексті впровадження комп'ютерного зору та глибокого навчання дозволяє автоматизувати критично важливі процеси, підвищити точність інвентаризації, скоротити час оброблення товарів та зменшити ризик помилок персоналу [1].

Розроблення інтелектуальної системи розпізнавання об'єктів на основі моделей комп'ютерного зору для автоматизації управління складськими запасами є актуальною науковою і прикладною проблемою, яка забезпечує модернізацію логістичних процесів, оптимізацію витрат підприємства та підвищення рівня надійності складської інфраструктури.

Метою дослідження є проєктування та створення інтелектуальної системи розпізнавання об'єктів, здатної забезпечувати автоматизований контроль складських запасів у режимі реального часу шляхом застосування сучасних алгоритмів комп'ютерного зору, інтегрованих зі складськими інформаційними системами.

Для досягнення поставленої мети сформульовано **такі основні завдання**:

– здійснити системний аналіз предметної області та визначити вимоги до програмно-апаратного комплексу;

- розробити архітектуру інтелектуальної системи з урахуванням специфіки складських операцій;
- обґрунтувати та вибрати моделі глибинного навчання для детекції, класифікації та підрахунку об'єктів;
- створити модулі збору, синхронізації та оброблення відеопотоків складської зони;
- реалізувати високопродуктивні алгоритми інференсу для роботи в умовах реального часу;
- розробити модуль інтеграції з WMS/ERP-системами та механізмами автоматизованого керування запасами;
- провести тестування точності, швидкодії та стійкості системи в умовах реальних сценаріїв складської логістики;
- оцінити ефективність упровадження системи у процеси інвентаризації, приймання, розміщення та відвантаження товарів.

Об'єктом дослідження є процес автоматизованого сприйняття та аналізу візуальної інформації у складських логістичних системах.

Предметом дослідження є методи, алгоритми та програмні засоби комп'ютерного зору, що забезпечують розпізнавання, класифікацію, підрахунок і відстеження товарних одиниць у режимі реального часу в умовах роботи складських комплексів.

Методологічною основою дослідження є методи глибинного навчання, згорткових і трансформерних архітектур, адаптивні алгоритми оптимізації інференсу на спеціалізованих апаратних засобах, методи статистичного аналізу достовірності оброблення даних, а також інструменти системного та структурного моделювання (UML, DFD, ER-моделювання).

Наукова новизна роботи полягає в розробленні цілісної інтегрованої системи, яка поєднує сучасні моделі комп'ютерного зору з логістичними механізмами управління запасами; у визначенні оптимальних параметрів інференсу для забезпечення високої точності розпізнавання за умов реального складського середовища; у розробленні архітектури, що забезпечує узгоджену

взаємодію візуальної аналітики, потоків операційної діяльності та складських інформаційних систем; а також у формуванні методики комплексного тестування ефективності системи в умовах змінних операційних навантажень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Сучасні складські комплекси характеризуються високим рівнем динамічності операцій, значними обсягами товарообігу та потребою в мінімізації людського фактору при виконанні логістичних процесів. У таких умовах критично важливими стають інтелектуальні системи комп'ютерного зору, здатні здійснювати автоматичне виявлення, класифікацію, кількісне оцінювання та відстеження товарів у режимі реального часу. На Рис. 1.1 подано узагальнену архітектуру потоків даних та взаємодії підсистем, що формують предметну область інтелектуального розпізнавання об'єктів для автоматизації управління складськими запасами.

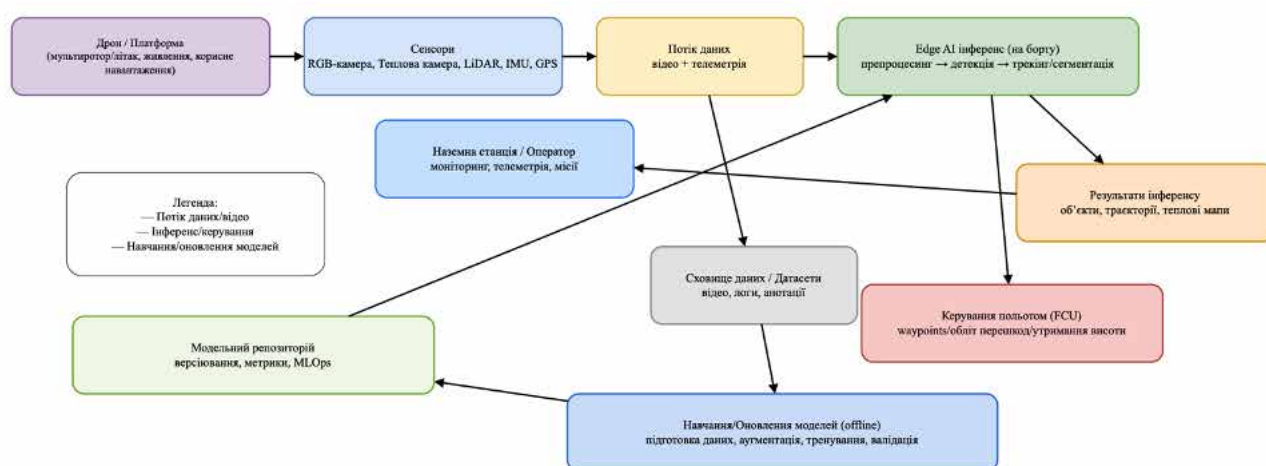


Рис. 1.1. Загальна архітектура потоків даних та компонентів системи розпізнавання об'єктів складського комплексу

Отримання вхідної інформації здійснюється системою технічного зору, до складу якої входять RGB-камери, глибинні камери, стереосенсори або систему відеоспостереження, інтегровані з локальними зонами приймання, зберігання та відвантаження товарів. Вони формують сировинні відеопотоки, що передаються до підсистеми Edge AI для подальшого препроцесування, нормалізації та виявлення об'єктів. У режимі реального часу тут відбувається детекція, класифікація, підрахунок одиниць продукції, контроль заповненості полиць, а

також ідентифікація аномальних подій (помилки комплектування, переплутана продукція, нерівномірне розміщення тощо). Швидкодія цієї підсистеми є визначальною для забезпечення автоматизації складських процесів, особливо в умовах великих товарних потоків та інтенсивності операцій.

На Рис. 1.2 наведено DFD-модель процесів збору відеоданих, інтелектуальної аналітики та формування управлінських дій у складській системі.

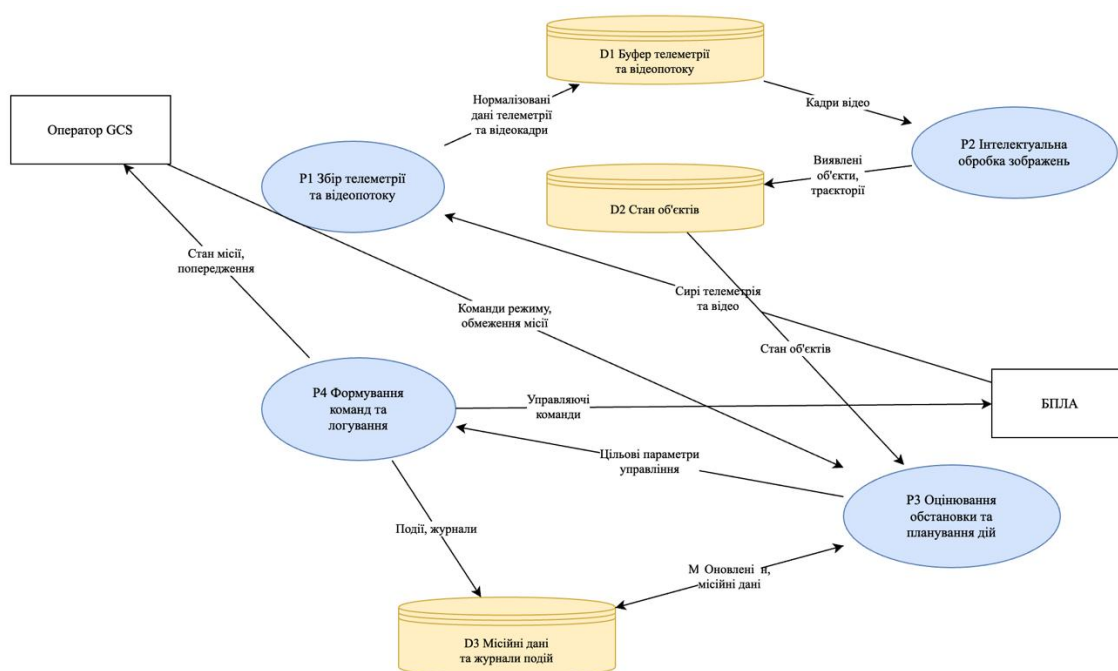


Рис. 1.2. DFD-модель процесів збору даних, інтелектуальної обробки та планування управлінських дій у складській логістиці

Після первинної обробки результати інференсу (класи об'єктів, їх кількість, координати розміщення, динаміка переміщень) інтегруються з контекстними даними складської інформаційної системи (WMS/ERP), що дозволяє формувати узагальнену модель стану запасів. Це забезпечує можливість автоматичного оновлення залишків, контролю відповідності продукції накладним, формування попереджень про дефіцит, виявлення надлишків та проведення автоматизованої інвентаризації. У разі застосування роботизованих платформ або автоматизованих конвеєрних систем передбачено зворотний зв'язок із виконавчими модулями, що дозволяє в реальному часі коригувати маршрути переміщення товарів.

Предметна область передбачає наявність централізованих або розподілених сховищ даних, що містять відеозаписи, метадані, логи, анотації та результати інтелектуального оброблення. Такі сховища виконують ключову роль у MLOps-циклі, зокрема у повторному тренуванні моделей, актуалізації датасетів, валідації нових версій та моніторингу продуктивності системи. Безперервний цикл збору даних, їх аналізу та подальшого вдосконалення моделей забезпечує зростання точності та надійності розпізнавання в умовах реальних складських процесів.

Для систем управління запасами характерні специфічні вимоги: забезпечення детекції товарних одиниць зі швидкістю не менше 20–30 FPS у режимах сортування й приймання, стійкість моделей до складних умов (часткові оклюзії, схожість упаковок, зміна ракурсів та освітлення), потреба в мінімальній затримці між інференсом та оновленням даних WMS, а також високий рівень точності при підрахунку об'єктів і виявленні невідповідностей.

У Таблиці 1.1 наведено узагальнені характеристики предметної області та ключові вимоги до системи інтелектуального розпізнавання об'єктів у контексті автоматизації складських запасів.

Таблиця 1.1.

Основні характеристики предметної області та вимоги до системи

| Компонент / Характеристика | Опис |
|----------------------------|---|
| Джерела даних | RGB-відео, глибинні камери, стереосенсори, відеосистеми на конвеєрах |
| Типи задач | Детекція, класифікація, підрахунок, трекінг, контроль заповненості |
| Обмеження | Продуктивність Edge-модулів, енергоспоживання, пропускна здатність мережі |
| Вимоги до швидкодії | 20–30 FPS для операцій приймання та сортування |
| Вимоги до точності | Мультикласове розпізнавання, коректний підрахунок товарів |
| Інтеграція | WMS, ERP, сенсорні системи складу, роботизовані платформи |
| Умови експлуатації | Зміни освітлення, щільне розміщення товарів, динамічні сцени |

Предметна область включає широкий спектр програмно-апаратних підсистем, інтеграція яких забезпечує повний цикл автоматизованого сприйняття, аналізу та прийняття рішень у складських комплексах. Інтелектуальна система розпізнавання об'єктів є центральним елементом цього циклу, визначаючи точність інвентаризації, правильність обліку, ефективність логістичних операцій та загальний рівень автоматизації управління запасами.

1.2 Теоретико-методологічні засади та стан наукових досліджень

Розвиток інтелектуальних систем розпізнавання об'єктів для складських підприємств ґрунтується на сучасних методах комп'ютерного зору, глибокого навчання та оптимізованих архітектурах штучних нейронних мереж. Основою таких систем є згорткові нейронні мережі (Convolutional Neural Networks, CNN), які демонструють високу точність у задачах детекції, трекінгу та сегментації об'єктів у складних візуальних умовах [1]. На Рис. 1.3 наведено базову двоетапну архітектуру детекції, у якій CNN використовується для формування пропозицій областей (proposal boxes), а на наступному етапі - для уточнення класифікації та локалізації.

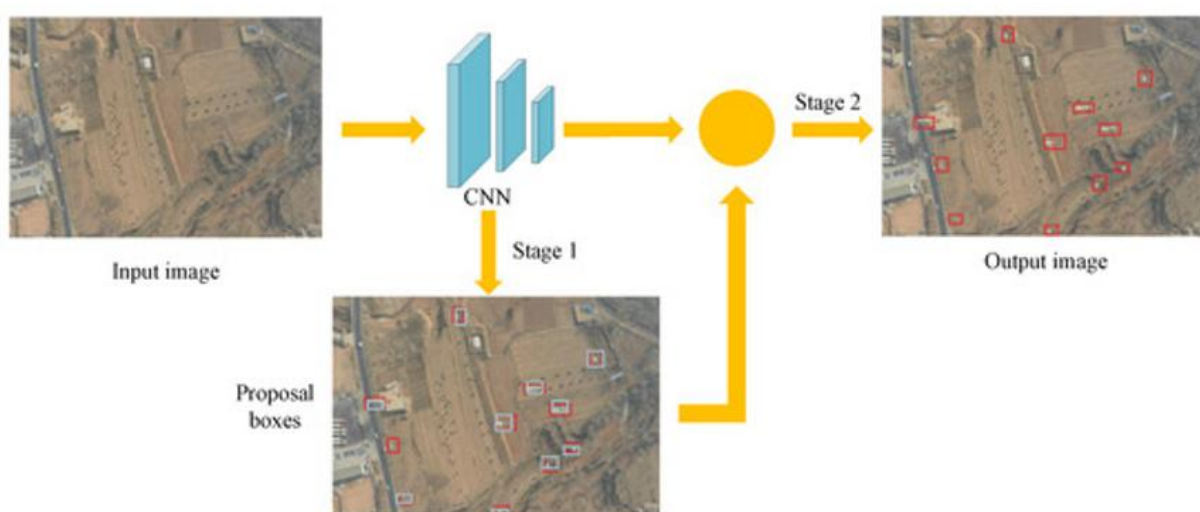


Рис. 1.3. Двоетапна модель детекції на основі CNN із генерацією областей-пропозицій

У наукових дослідженнях значну увагу приділено одноетапним детекторам, що оптимізують швидкодію завдяки прямому прогнозуванню координат об'єктів без проміжних стадій. Найпоширенішими є моделі сімейств YOLO (You Only Look Once) та SSD (Single Shot Detector). Згідно з роботами Редмонда та Фарбера [2], YOLO забезпечує високу швидкість обробки за рахунок єдиного проходження зображення через CNN, що робить модель придатною для застосування на борту БПЛА. SSD, запропонована Лю та колегами [3], використовує багатомасштабні фічі для підвищення стійкості до різних розмірів об'єктів. На Рис. 1.4 зображено порівняльну схему роботи YOLO та SSD, що демонструє їхні концептуальні відмінності у формуванні вихідних рамок.

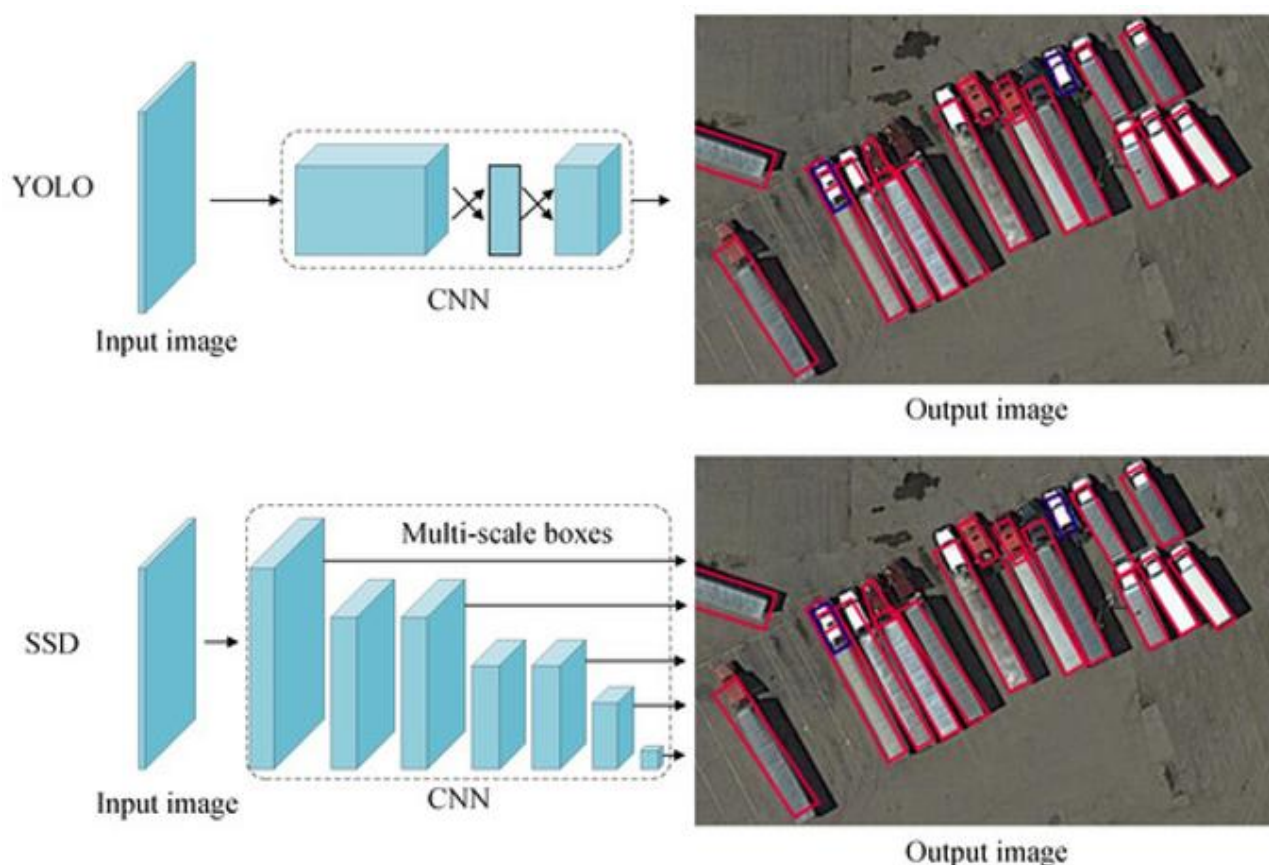


Рис. 1.4. Одноетапні детектори YOLO та SSD: архітектурні принципи

Поступальний розвиток методів розпізнавання зумовлений переходом до механізмів уваги (Attention), які було вперше системно описано Васвані та співавторами [4]. Завдяки можливості моделювати залежності між віддаленими пікселями механізми уваги дозволяють підвищити точність розпізнавання у

сценах зі складною текстурою й частковими оклюзіями. У роботах 2021–2024 рр. увагу інтегрують у CNN-архітектури через блоки self-attention, двовимірне згортання та крос-модальні механізми. На Рис. 1.5 показано типовий приклад застосування двовимірної уваги у детекції, де карти Q, K та V формуються окремими CNN-перетвореннями та комбінуються за допомогою softmax-зважування.

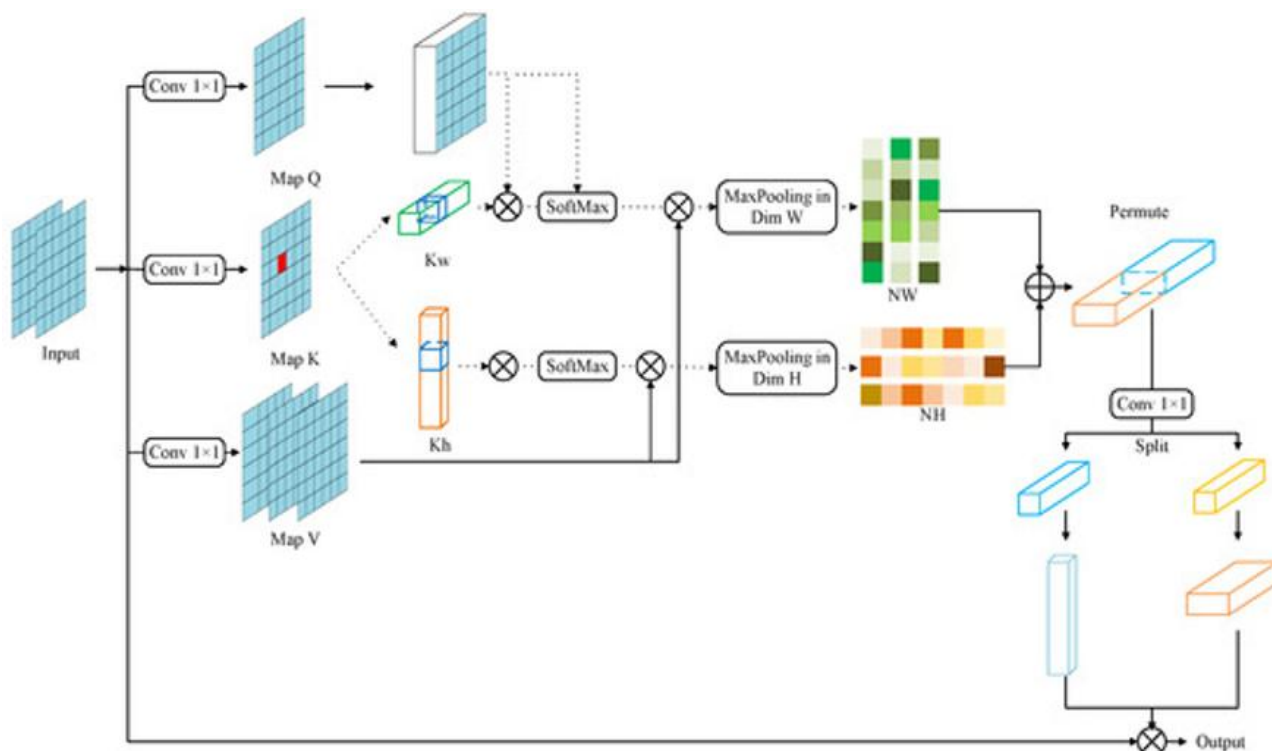


Рис. 1.5. Архітектура блоку 2D-attention для покращення детекції об'єктів

Паралельно розвивалися двоетапні моделі сімейства R-CNN (Fast R-CNN, Faster R-CNN, Mask R-CNN), які забезпечують високу точність завдяки регіональним пропозиціям, сформованим за допомогою Region Proposal Network (RPN). Дослідження Хе та співавторів [5] довели ефективність таких моделей у задачах з високою щільністю об'єктів та складною геометрією сцени. На Рис. 1.6 зображено узагальнену архітектуру Faster R-CNN, що включає

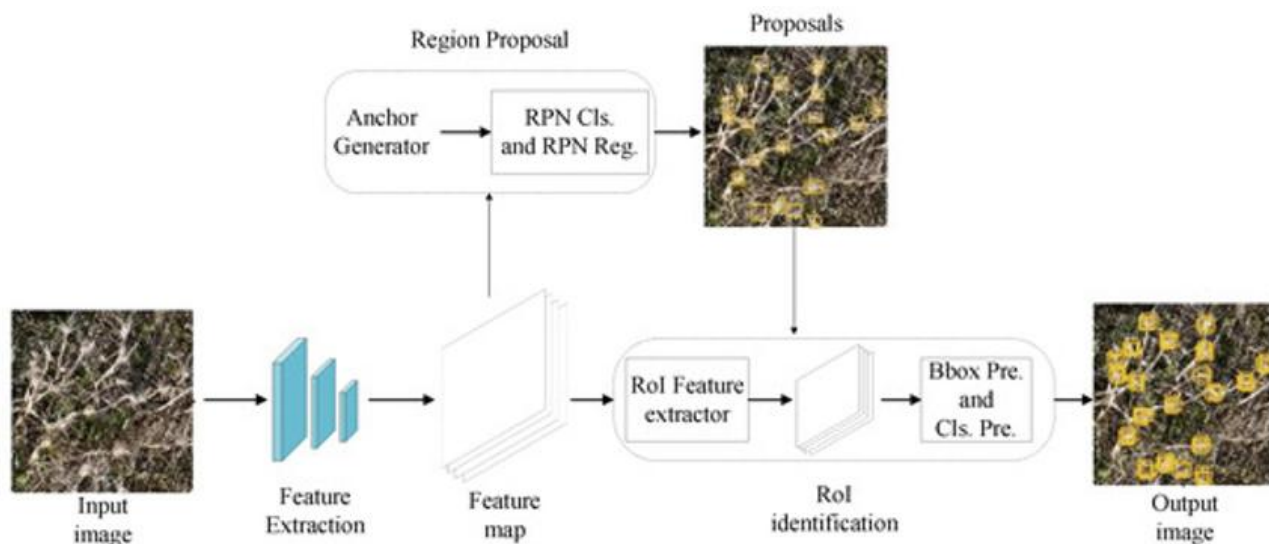


Рис. 1.6. Узагальнена схема Faster R-CNN для детекції об'єктів у складних сценах

У практичних застосуваннях інтелектуальних систем складських підприємств часто постає потреба в поєднанні детекції, трекінгу та сегментації об'єктів. Використання сегментаційних моделей (Mask R-CNN, DeepLab, U-Net) дозволяє виділяти структуру об'єктів у просторі, що є корисним для аграрного моніторингу, технічного огляду та пошукових задач. На Рис. 1.7 наведено приклад поєднання сегментації та детекції для точної локалізації об'єктів складної форми.

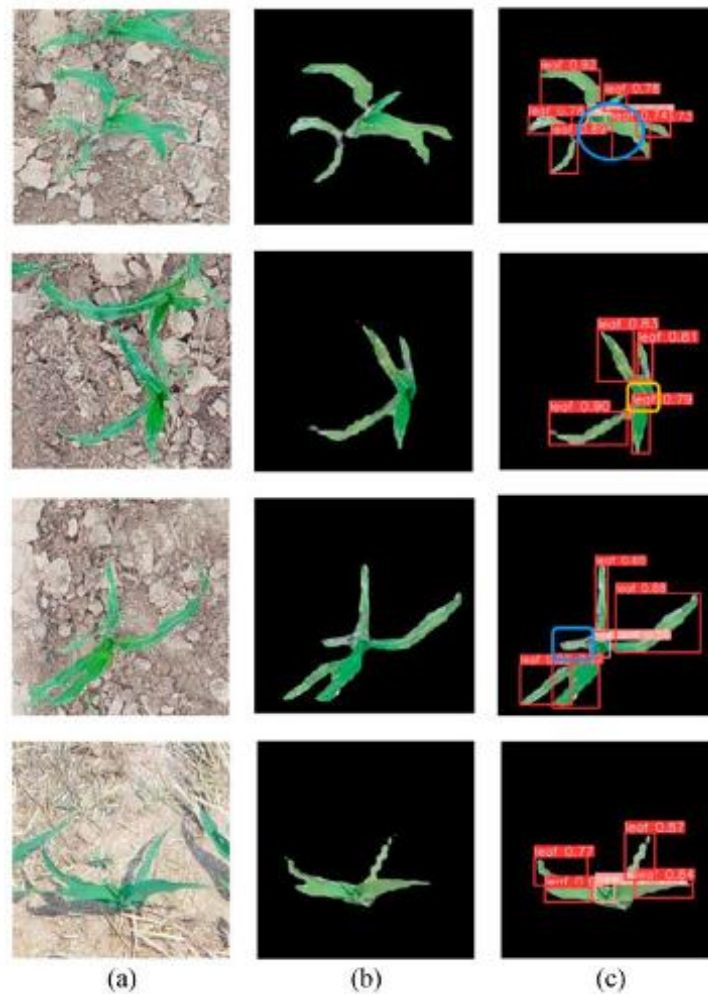


Рис. 1.7. Приклад комбінованої сегментації та детекції об'єктів

У наукових дослідженнях останніх років активно вивчають інтеграцію глибинних моделей із контуром управління БПЛА. Роботи Шенга, Лі та Сунь [6] підкреслюють важливість низької затримки інференсу та стійкості моделей до шумових артефактів у реальному польоті. Проблеми узгодження даних сенсорів, телеметрії та візуальних ознак розв'язуються шляхом багатомодальних моделей, які поєднують RGB-відео, глибинні карти та IMU-сигнали.

Системи MLOps також відіграють ключову роль у вдосконаленні моделей, забезпечуючи генерацію нових датасетів, повторне тренування, моніторинг метрик і керування версіями. Автоматизоване оновлення моделей дозволяє підтримувати їхню актуальність у змінних умовах середовища, що є критично важливим для складських підприємств, які працюють у різних часових та світлових сценаріях.

Аналіз сучасних підходів засвідчує, що хоча існуючі нейромережеві моделі забезпечують високу точність у задачах детекції та сегментації, їх застосування потребує оптимізації інференсу, зменшення обчислювальної складності та стійкості до шумових перешкод. Наукова новизна даної роботи полягає у поєднанні механізмів уваги з оптимізованими CNN-архітектурами для підвищення точності детекції в режимі реального часу, інтеграції інтелектуальної обробки з телеметричними потоками та розробленні узгодженої архітектури, що забезпечує стабільну взаємодію між інференсом і контуром керування. Крім того, застосування підходів MLOps дозволяє формувати замкнений цикл адаптивного вдосконалення моделей, що підвищує надійність та ефективність системи у реальних умовах експлуатації.

1.3 Аналіз існуючих рішень

Сучасні програмно-апаратні комплекси, що застосовуються у складській логістиці, інтегрують модулі відстеження товарних потоків, збору відеоданих, автоматизованої інвентаризації, аналітики руху об'єктів, прогнозування залишків та керування роботизованими платформами. У межах даного підрозділу розглянуто найпоширеніші системи комп'ютерного зору, WMS-рішення та роботизовані комплекси, що використовуються на складах для контролю товарів, з урахуванням їх можливостей, обмежень і відповідності вимогам інтелектуального розпізнавання об'єктів у режимі реального часу.

Першу групу рішень формують системи комп'ютерного зору для підрахунку, ідентифікації та контролю товарних одиниць (Amazon Kiva Vision, Zebra SmartPack, Bosch DeepField). На Рис. 1.8 представлено типовий приклад роботи системи SmartPack: на основі відеопотоку формується просторове представлення вантажів, обчислюються їх габарити та контролюється завантаженість платформи. Такі системи демонструють високу точність виявлення об'єктів, але здебільшого орієнтовані на контроль вантажних

операцій і не передбачають глибокої інтеграції з бортовими AI-модулями або сценарними алгоритмами оптимізації складських запасів.

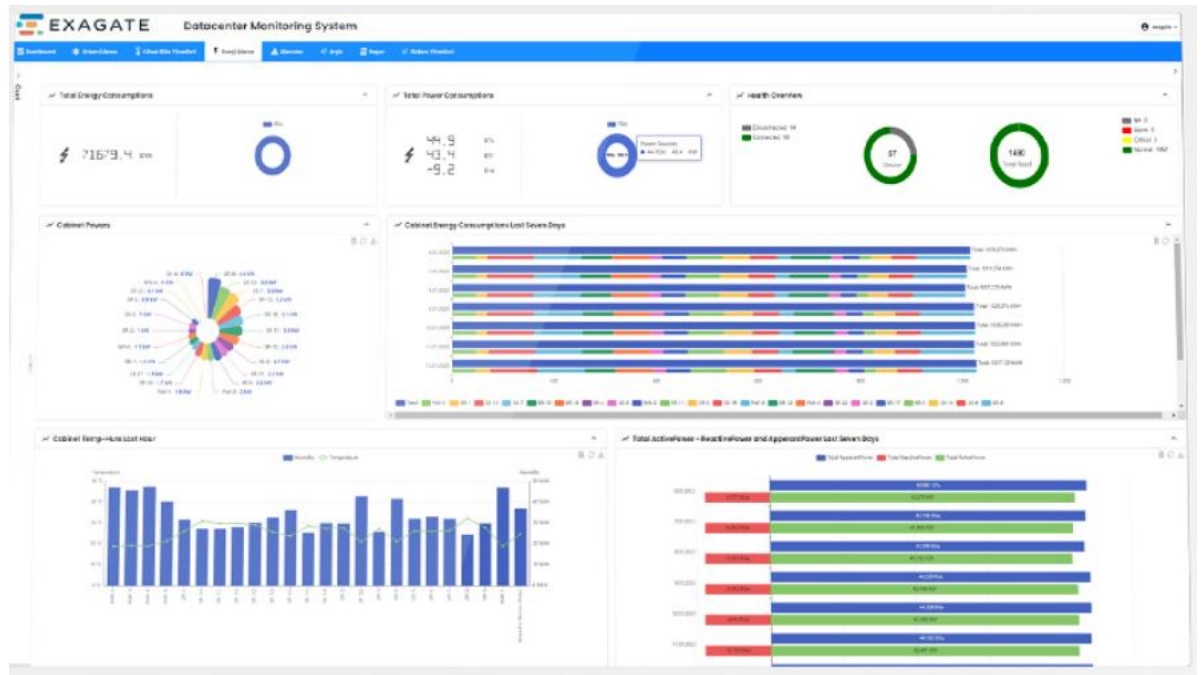


Рис. 1.8. Контроль товарних потоків і підрахунок вантажів у системі SmartPack

Другу групу складають WMS-платформи з інструментами візуального моніторингу (Oracle WMS Cloud, SAP EWM, Manhattan Active Warehouse Management). На Рис. 1.9 показано приклад інтерфейсу SAP EWM, який надає можливості відстеження руху товарів, керування зоною зберігання та оброблення замовлень. Такі системи забезпечують потужні механізми управління внутрішніми операціями, проте зазвичай не містять вбудованого модулю комп'ютерного зору для автоматичного розпізнавання продукції, покладаючись на ручне введення або RFID-ідентифікацію.

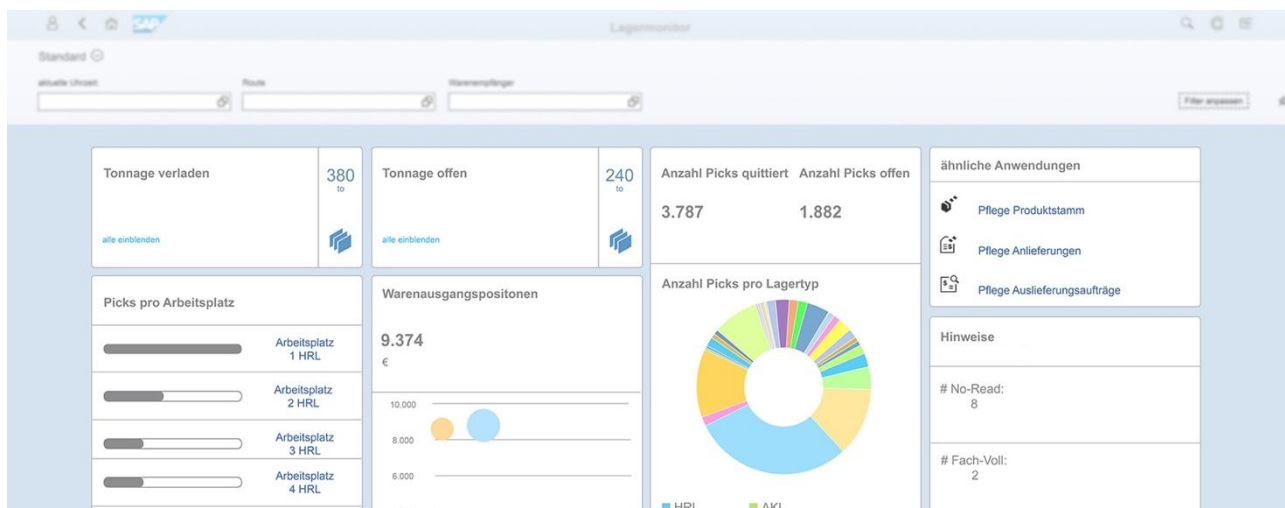


Рис. 1.9. Інтерфейс SAP EWM з візуалізацією складських процесів

До третьої групи належать роботизовані платформи та автономні мобільні роботи (AMR), такі як системи Kiva Robotics (Amazon), GreyOrange Butler та Нікrobot FMR. На Рис. 1.10 наведено приклад інтерфейсу системи Kiva Robotics, де відображено маршрути, статуси роботів та динаміку виконання завдань. Хоча такі рішення значно підвищують автоматизацію, більшість із них не мають власних модулів комп'ютерного зору для розпізнавання товарів, покладаючись на QR-мітки, штрихкоди або попередньо визначені структуровані зони.



Рис. 1.10. Інтерфейс роботизованої платформи Kiva Robotics

Окремий клас рішень становлять системи багатосенсорного контролю та автоматизованого сортування (Siemens Multiscan, Honeywell Intelligrated), що поєднують RGB-камери, глибинні сенсори та конвеєрні лінії. На Рис. 1.11

зображено типовий сценарій, коли RGB-камери та лазерні сканери одночасно знімають вантажі для визначення габаритів і перевірки правильності сортування. Такі платформи здатні працювати у високонавантажених середовищах, але не виконують глибокий аналіз складських запасів (інвентаризація, відповідність номенклатурі, контроль дефіциту).



Рис. 1.11. Мультисенсорна система контролю на конвеєрній лінії

Додатково значне поширення отримали фотограмметричні та сканувальні системи для побудови 3D-моделей складів (Matterport Pro3, NavVis VLX). На Рис. 1.12 представлено типовий результат 3D-сканування складського приміщення, що використовується для інвентаризації, побудови цифрового двійника або аналізу ефективності просторового розміщення. Проте такі системи придатні лише для постобробки, а не для автоматичного онлайн-розпізнавання товарів.

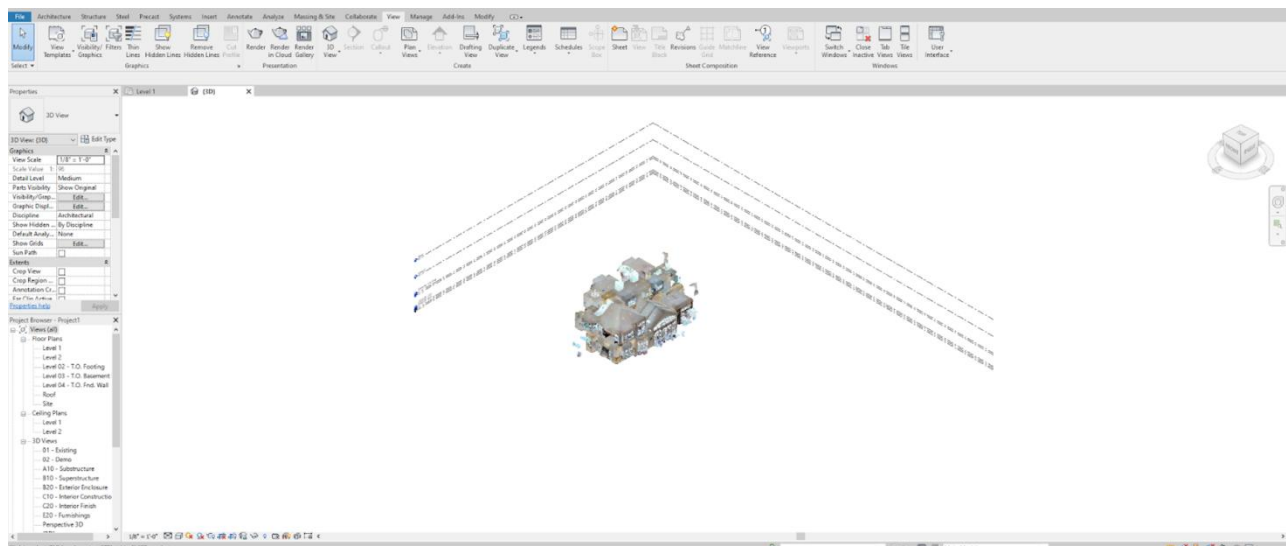


Рис. 1.12. 3D-моделювання складського простору у Matterport

Для узагальнення аналізу у Таблиці 1.2 наведено порівняння ключових характеристик розглянутих систем та розроблюваної інтелектуальної системи, що поєднує розпізнавання, трекінг, підрахунок, теплову та глибинну аналітику та повноцінну інтеграцію з WMS/ERP-рішеннями.

Таблиця 1.2

Порівняльний аналіз аналогів і розроблюваної системи

| Система | Тип | Детекція в реальному часі | Трекінг | 3D-моделювання | Інтеграція з WMS/ERP | AI-інференс (Edge) | Призначення |
|----------------------|---------------------|---------------------------|----------|----------------|----------------------|--------------------|--------------------|
| Amazon Kiva Vision | Комп'ютерний зір | Частково | Так | Ні | Так | Ні | Контроль AMR |
| Zebra SmartPack | Візуальна аналітика | Так | Частково | Частково | Ні | Ні | Аналіз потоків |
| SAP EWM | WMS | Ні | Ні | Ні | Так | Ні | Управління складом |
| Honeywell Integrated | Сортування | Так | Ні | Частково | Частково | Ні | Конвеєри |
| Matterport Pro3 | 3D-сканування | Ні | Ні | Так | Частково | Ні | Постобробка |

Продовження таблиці 1.2

| | | | | | | | |
|-----------------------------------|---------------------|--|-----|------------------------|-----------------------------|---------------|-----------------------------------|
| Розроблювана система (пропозиція) | AI-контроль запасів | Так (детекція + підрахунок + класифікація) | Так | Так (цифровий двійник) | Так (двостороння взаємодія) | Так (Edge AI) | Автоматизація управління запасами |
|-----------------------------------|---------------------|--|-----|------------------------|-----------------------------|---------------|-----------------------------------|

Проведений аналіз показує, що наявні ринкові рішення здебільшого орієнтовані на окремі складські процеси - сортування, сканування, роботизацію або управління операціями – але не забезпечують інтегрованої інтелектуальної системи, здатної:

- автоматично розпізнавати товарні одиниці (мультикласово);
- виконувати підрахунок, контроль заповненості, виявлення аномалій;
- синхронізувати дані з WMS/ERP;
- аналізувати 3D-структуру складського простору;
- формувати автоматизовані управлінські команди для AMR-платформ;
- працювати на Edge-обладнанні в режимі реального часу.

Таким чином, існує суттєва потреба у створенні комплексної інтелектуальної системи розпізнавання об'єктів, яка забезпечить повний цикл автоматизації управління складськими запасами, підвищення точності інвентаризації, мінімізацію людського фактору та оптимізацію внутрішніх логістичних процесів.

1.4 Моделювання предметної області

Моделювання предметної області інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими підприємствами базується на формалізації взаємодії акторів, програмних компонентів, потоків даних та алгоритмічних процесів. Застосування UML-діаграм дозволяє структурувати функціональну логіку системи, відобразити поведінку підсистем, а також

описати ключові сценарії взаємодії у контексті роботи емулятора, підсистеми інференсу та контурів управління.

Першим елементом моделювання виступає діаграма послідовності, яка описує роботу підсистеми автентифікації, перевірки доступу та подальшої взаємодії користувача із захищеними ресурсами. На Рис. 1.13 продемонстровано послідовність операцій між актором, вебклієнтом, сервісом автентифікації, MFA-модулем, політиками доступу та системою журналювання. Це моделювання є необхідним, оскільки робочий процес емульованої системи розпізнавання об'єктів також базується на безпечному доступі до моделей, даних та сервісів телеметрії.

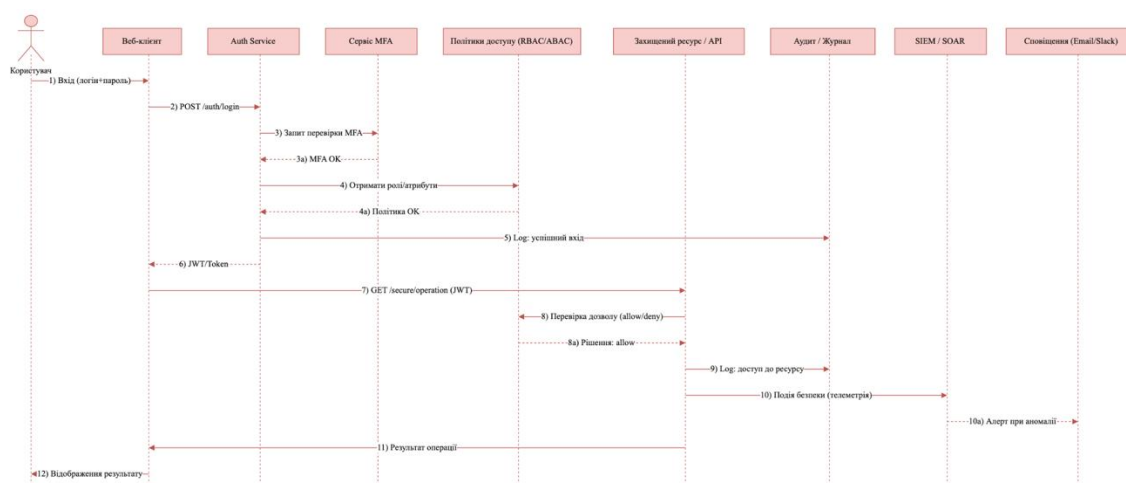


Рис. 1.13. UML-діаграма послідовності авторизації та взаємодії з підсистемами доступу

Далі, для опису функціональних можливостей системи побудовано діаграму прецедентів, що визначає ролі акторів та їхні взаємодії із програмним емулятором. На Рис. 1.14 наведено відображення основних сценаріїв: запуск та зупинка симуляції, моніторинг детекцій, аналіз продуктивності моделей, оновлення та завантаження моделей через MLOps, збереження логів і датасетів. Актори «Оператор», «Аналітик» та «MLOps сервіс» взаємодіють із системою у різних контекстах, що забезпечує масштабованість та модульність архітектури.



Рис. 1.14. Діаграма прецедентів програмного емулятора системи розпізнавання об'єктів для дронів

Третім елементом моделювання є діаграма активності, що описує алгоритм роботи програмного емулятора, починаючи від завантаження конфігурації та параметрів сценарію до формування команд FCU та логування. На Рис. 1.15 відображено повний цикл обробки: генерація синтетичних сцен або отримання реальних сенсорних даних, препроцесинг відеопотоку, виконання ML-інференсу на основі моделей YOLO/SSD, трекінг об'єктів, оцінка траєкторій, формування керуючих команд і завершення симуляції зі збереженням метрик.

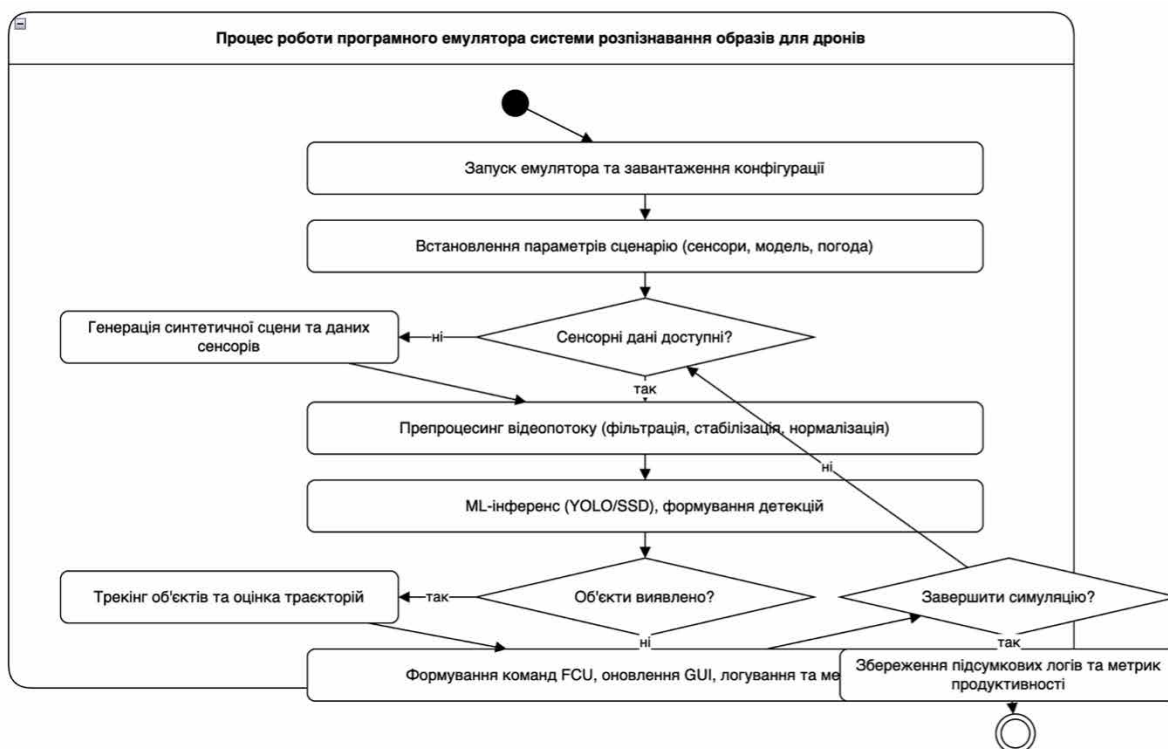


Рис. 1.15. UML-діаграма активності процесу роботи інтелектуального емулятора системи розпізнавання об'єктів

Узагальнюючи результати моделювання, можна зазначити, що інтеграція діаграм послідовності, прецедентів та активності дозволяє сформуванню цілісного уявлення про поведінку системи та її структурне наповнення. Моделі демонструють, що система має модульну архітектуру з чітким поділом відповідальностей: підсистема інференсу відповідає за детекцію та оцінку траєкторій; підсистема симуляції забезпечує відтворення сценаріїв та обробку відеопотоку; підсистема управління забезпечує взаємодію з FCU; а підсистема MLOps - оновлення та версіонування моделей. Таким чином, моделювання предметної області підтверджує, що запропонована система може бути ефективно реалізована у вигляді комплексного програмного емулятора з підтримкою реальних та синтетичних даних, забезпечуючи високу модульність, масштабованість та інтеграцію з сучасними алгоритмами комп'ютерного зору.

1.5 Аналіз вимог інформаційної системи

Аналіз вимог інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими запасами спрямований на формальне визначення функціональних, нефункціональних та безпекових характеристик, які забезпечують точність, надійність і стабільність роботи системи в умовах динамічних складських процесів. Вимоги сформовано на основі моделювання предметної області, аналізу існуючих рішень, теоретико-методологічних засад комп'ютерного зору та специфіки роботи високонавантажених складських комплексів.

Функціональні вимоги визначають основні можливості системи, пов'язані з обробленням відеопотоку, розпізнаванням об'єктів, підрахунком товарних одиниць, контролем складських зон та інтеграцією з WMS/ERP. У Таблиці 1.3 узагальнено ключові функції системи, які забезпечують повний цикл автоматизованого аналізу складських запасів.

Таблиця 1.3

Функціональні вимоги інтелектуальної системи

| № | Вимога | Опис |
|----|---------------------------|---|
| F1 | Оброблення відеопотоку | Прийом RGB/Depth-відео зі складу, стабілізація, нормалізація та препроцесинг |
| F2 | Детекція товарних одиниць | Використання моделей YOLO/RetinaNet для виявлення продукції різних класів у реальному часі |
| F3 | Підрахунок і класифікація | Визначення кількості товарів, розпізнавання типу, упаковки, групи зберігання |
| F4 | Трекінг переміщень | Відстеження руху товарів, коробів, палет, формування траєкторій на конвеєрах і в зонах комплектування |
| F5 | Контроль складських зон | Виявлення заповненості полиць, перевантаження стелажів, невідповідностей розміщення |
| F6 | Інтеграція з WMS/ERP | Передача даних про залишки, розміщення, інвентаризаційні відхилення, оновлення SKU-позицій |
| F7 | Взаємодія з GUI | Відображення детекцій, підрахунку, зон ризику, індикаторів заповненості та стану системи |
| F8 | Логування та збір даних | Запис метрик, відеофреймів, журналів подій, збереження датасетів для MLOps |

Продовження таблиці 1.3

| | | |
|----|---------------------|---|
| F9 | Підтримка симуляції | Генерування синтетичних сценаріїв, тестування моделей у віртуальному складському середовищі |
|----|---------------------|---|

Нефункціональні вимоги визначають якісні параметри системи, включаючи продуктивність, масштабованість, відмовостійкість, точність розпізнавання та сумісність з апаратним забезпеченням. Усі ці характеристики є критичними для роботи в режимі реального часу у складських комплексах, де обсяг операцій може досягати десятків тисяч товарних одиниць на годину.

Таблиця 1.4

Нефункціональні вимоги інтелектуальної системи

| № | Вимога | Опис |
|-----|---------------------|--|
| NF1 | Швидкодія інференсу | $\geq 20\text{--}30$ FPS для стабільної роботи інвентаризаційних і сортувальних процесів |
| NF2 | Затримка обробки | $\leq 120\text{--}150$ мс між надходженням кадру та формуванням детекцій |
| NF3 | Надійність | Безперервна робота $\geq 99\%$ часу, підтримка автономної діагностики |
| NF4 | Масштабованість | Підтримка різних моделей (YOLOv8, EfficientDet, MobileNet) і сценаріїв роботи складу |
| NF5 | Портативність | Робота на Edge-платформах (NVIDIA Jetson, Intel Movidius, ARM-комплекси) |
| NF6 | Якість підрахунку | Похибка підрахунку $\leq 5\%$ навіть у випадках часткових оклюзій |
| NF7 | Відмовостійкість | Автоматичне відновлення після втрати кадрів, мережевих збоїв, локальних помилок |
| NF8 | Інтероперабельність | Підтримка REST API, gRPC, ONNX Runtime, інтеграція з WMS/ERP, AMR-системами |

Оскільки система обробляє відеодані, інформацію про запаси та результати інвентаризації, що мають комерційну та операційну цінність, критично важливим є забезпечення високого рівня кібербезпеки, захисту даних і контроль доступу. У Таблиці 1.5 наведено основні вимоги до безпеки системи.

Таблиця 1.5

Вимоги до безпеки інтелектуальної системи

| № | Вимога | Опис |
|----|-----------------------------|---|
| S1 | Автентифікація користувачів | Підтримка MFA, токенів доступу, ролей для персоналу складу та адміністраторів |
| S2 | Контроль доступу | RBAC/ABAC-моделі для обмеження дій операторів і сервісів |
| S3 | Захищені канали | TLS 1.3/HTTPS для передавання відео, журналів, API-викликів |
| S4 | Верифікація моделей | Перевірка цілісності моделей, контроль оновлень, захист від компрометації |
| S5 | Безпечний інференс | Обмеження доступу до критичних модулів WMS та інвентаризації |
| S6 | Виявлення аномалій | Аналіз метрик і відеоданих для виявлення підозрілих операцій, крадіжок, втрат |
| S7 | Захист логів | Шифрування журналів, контроль доступу до MLOps-середовища |
| S8 | Інцидент-менеджмент | Інтеграція із SIEM/SOAR для автоматичного формування попереджень |

Виконаний аналіз вимог демонструє, що система повинна забезпечувати комплексний функціонал для роботи з багатомодальними потоками даних, виконання інференсу моделей комп'ютерного зору, підрахунку та контролю товарів, інтеграції з WMS/ERP, підтримки симуляцій і реалізації глибокого рівня безпеки. Високі вимоги до швидкодії, точності та надійності визначають необхідність оптимізації ML-моделей для Edge-платформ, використання стійких алгоритмів трекінгу та побудови масштабованої мікросервісної архітектури. Наявність розширених безпекових вимог підтверджує важливість багаторівневого підходу до захисту, що охоплює контроль доступу, захищені канали, аудит, логування та моніторинг інцидентів. Таким чином, сформовані вимоги забезпечують комплексну основу для побудови інтелектуальної системи управління складськими запасами, здатної працювати в реальному часі, гарантувати точність інвентаризації, оптимізувати логістичні процеси та підвищувати загальну ефективність складської інфраструктури.

1.6 Постановка завдання

Постановка завдання для розроблення інтелектуальної системи розпізнавання об'єктів, призначеної для автоматизації управління складськими запасами, ґрунтується на сформульованих вимогах, описі предметної області та аналізі сучасних підходів комп'ютерного зору. Основна мета полягає у створенні програмно-алгоритмічного комплексу, здатного в режимі реального часу обробляти відеопотоки зі складської інфраструктури, виконувати інференс оптимізованих моделей глибинного навчання, здійснювати підрахунок та класифікацію товарних одиниць, формувати аналітичну оцінку поточного стану зони зберігання та забезпечувати автоматизовану взаємодію з WMS/ERP-системами для підтримки процесів інвентаризації, контролю залишків і оптимізації логістичних операцій.

У контексті цього завдання система повинна приймати як вхідні дані поточкові RGB або глибинні відеодані, показники сенсорів конвеєрних ліній, інформацію про положення товарів у зоні сканування, контекстні дані WMS, конфігураційні параметри інвентаризаційних сценаріїв, а також моделі штучного інтелекту, завантажені з MLOps-сервісу або локального репозиторію. Вхідний масив даних охоплює кадри відеопотоку з різних зон складу, метадані про SKU, характеристики стелажів, параметри камер, журнали складських операцій, а також попередні результати детекції та трекінгу, що дозволяє виконувати багаторівневий аналіз стану запасів. Система має забезпечити повний цикл перетворення цих даних, включаючи стабілізацію зображень, нормалізацію освітлення, виділення ознак, детекцію та трекінг товарних одиниць, контроль заповненості полиць, виявлення аномалій у розміщенні, оцінку змін залишків і узгодження результатів з інформаційною моделлю складу.

Вихідними даними системи є структуровані результати інтелектуального аналізу, що містять координати та класи товарів, їх кількість, траєкторії

переміщень у зонах сортування або комплектування, оцінки правильності розміщення, показники щільності заповнення полиць, прогнозні значення зміни залишків, а також автоматично сформовані запити до WMS/ERP щодо оновлення інвентаризаційних даних, закриття складських операцій або ініціювання коригувальних дій. Додатковими вихідними даними є журнали подій, метрики продуктивності моделей, збережені датасети для повторного навчання, результати симуляцій і дані MLOps-моніторингу, що забезпечують замкнений цикл вдосконалення системи.

З урахуванням вимог до швидкодії, точності та безперервності роботи система повинна реалізовувати архітектуру, здатну функціонувати на ресурсно обмежених Edge-платформах, забезпечувати інтеграцію різнорідних сенсорних потоків, узгоджувати детекції з інформаційними даними WMS та забезпечувати прогнозовану поведінку у складських процесах із високою динамікою. Сукупність вхідних та вихідних даних формує цілісну логіку побудови інтелектуальної системи, яка здатна виконувати автоматизовану інвентаризацію, контроль залишків, підвищувати ефективність складських операцій і забезпечувати точність прийняття рішень у режимі реального часу.

1.7 Висновки до першого розділу

У першому розділі було здійснено комплексний системний аналіз предметної області інтелектуального розпізнавання об'єктів для автоматизації управління складськими запасами, що дало змогу узгодити концептуальні засади, визначити ключові проблеми сучасних складських систем і сформулювати вимоги до майбутнього програмно-алгоритмічного комплексу. На основі огляду наукових досліджень встановлено, що сучасні моделі комп'ютерного зору — від згорткових нейронних мереж до трансформерних архітектур — забезпечують високоточне розпізнавання товарних одиниць навіть за умов оклюзій, неоднорідного освітлення, щільного розміщення продукції та динамічного руху

товарних потоків, що є критично важливим для точності інвентаризації та оптимізації логістичних операцій у високонавантажених складських комплексах.

Аналіз існуючих рішень продемонстрував, що хоча сучасні WMS-платформи, роботизовані системи та інструменти комп'ютерного зору (Amazon Kiva Vision, Zebra SmartPack, Honeywell Intelligated, SAP EWM тощо) досягли значного прогресу в автоматизації операцій, вони не забезпечують комплексної інтеграції мультимодального відеоаналізу, підрахунку, трекінгу, цифрових двійників, аналітики заповненості й автоматичного оновлення залишків у WMS у режимі реального часу. Відсутність єдиного інтелектуального ядра, здатного обробляти дані на Edge-платформах і забезпечувати повний цикл «детекція → трекінг → аналіз → оновлення запасів», підтверджує потребу у створенні більш гнучкої, адаптивної та глибоко інтегрованої системи.

Моделювання предметної області за допомогою UML-діаграм дозволило чітко визначити ролі користувачів, сценарії оброблення відеоданих, структуру потоків інформації, взаємодію між підсистемами детекції, трекінгу, аналітики, WMS-інтеграції та MLOps. Це забезпечило формування логічної основи архітектури розроблюваної інтелектуальної системи та дало змогу описати повний цикл її функціонування в контексті складської логістики.

Аналіз вимог визначив набір функціональних, нефункціональних і безпекових характеристик, що охоплюють вимоги до швидкодії інференсу, точності підрахунку, відмовостійкості, масштабованості, інтеграції з WMS/ERP, безпеки даних, роботи на ресурсно обмежених Edge-платформах і підтримки різних складських сценаріїв. Усі ці вимоги формують технічну основу системи та забезпечують її здатність працювати у реальному часі.

Сформульована постановка завдання узагальнила результати проведеного аналізу, визначивши вхідні та вихідні дані, логіку роботи системи, необхідність багаторівневої обробки відеопотоків, інтеграції з контекстними даними WMS, використання моделі цифрового двійника простору, підтримки циклу MLOps та забезпечення автоматизованого оновлення запасів. Таким чином, перший розділ закладає науково-методологічну основу для подальшого проектування

архітектури системи, розроблення алгоритмів розпізнавання та трекінгу, побудови модулів інтелектуального аналізу складських процесів і створення комплексного програмно-апаратного рішення для високоточної автоматизації управління складськими запасами в режимі реального часу.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних розпізнавання об'єктів складських підприємств

Логічна модель даних формує структурну основу інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими підприємствами, оскільки саме узгоджене подання сутностей, атрибутів та зв'язків забезпечує коректність оброблення телеметрії, відеопотоків, результатів інференсу та метаданих моделей. На рис. 2.1 подано ER-діаграму, що відображає ключові сутності предметної області: платформу дрона, місію польоту, телеметричні записи, відеокадри, детекції та версії моделей комп'ютерного зору.

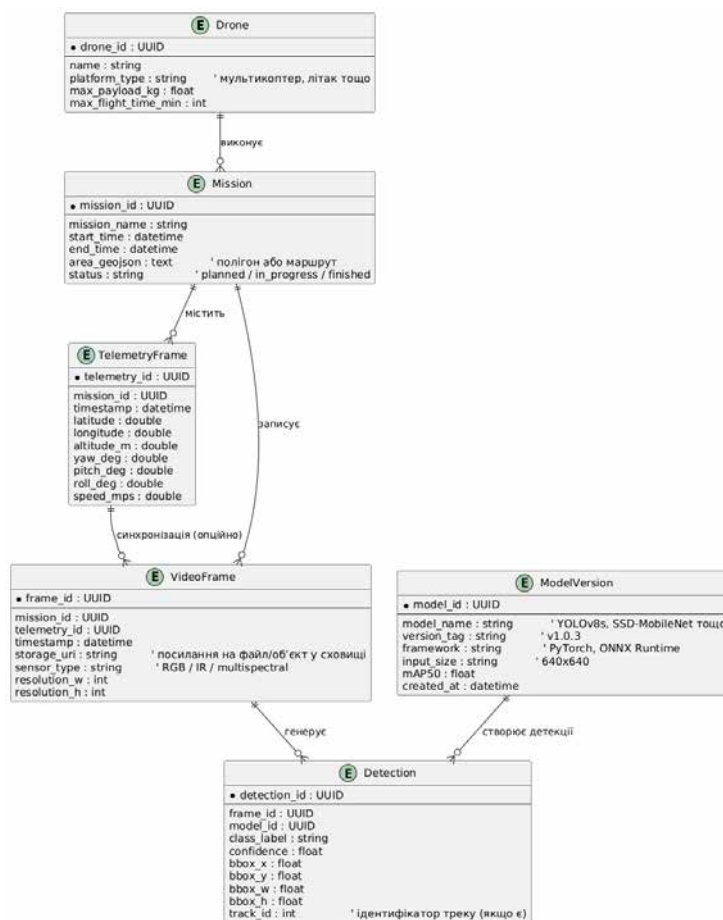


Рис. 2.1. Логічна модель даних інтелектуальної системи розпізнавання об'єктів складських підприємств у вигляді ER-діаграми

Побудована модель є результатом нормалізації даних до третьої нормальної форми, що дає змогу усунути дублювання інформації (наприклад, телеметричних параметрів у відеокадрах), мінімізувати аномалії оновлення та підвищити узгодженість усіх елементів системи. Розділення сутностей Mission, TelemetryFrame та VideoFrame з окремими первинними ключами гарантує можливість горизонтального масштабування для оброблення великих потоків даних, притаманних роботі БПЛА у реальних сценаріях. Окрема сутність ModelVersion дає змогу формувати централізований реєстр моделей комп'ютерного зору, підтримувати MLOps-процеси, забезпечувати відтворюваність інференсу та коректне порівняння продуктивності моделей.

Для повноцінної валідації схеми було сформовано узагальнену логічну таблицю компонентів даних, наведену у табл. 2.1, що відображає ключові сутності та їх роль у моделі.

Таблиця 2.1

Основні сутності логічної моделі даних та їх призначення

| Сутність | Основне призначення |
|----------------|--|
| Drone | Ідентифікація платформи, узгодження параметрів польоту та навантаження під час симуляцій і реальних місій. |
| Mission | Формалізація структурованої місії польоту, маршруту та часових рамок виконання. |
| TelemetryFrame | Зберігання параметрів орієнтації, швидкості та положення дрона для подальшої синхронізації та аналізу. |
| VideoFrame | Представлення окремого кадру відеопотоку, прив'язаного до телеметрії та місії. |
| Detection | Зберігання результатів ML-інференсу, включно з bounding-box, класами та довірчими ймовірностями. |
| ModelVersion | Версіонування та атрибутивний опис моделей комп'ютерного зору (YOLO, SSD), необхідних для відтворюваності результатів. |

Узагальнюючи, логічна модель даних забезпечує концептуальну цілісність інтелектуальної системи, оскільки відображає повний шлях даних — від захоплення сенсорної інформації та відеопотоків до формування детекцій і трекінгу об'єктів у реальному часі. Важливим результатом моделювання є те, що структура даних адаптована до подальшого розширення під потреби симулятора,

MLOps-інструментів та автономного управління БПЛА для роботи на складських підприємствах, а також передбачає інтеграцію потокової обробки і масштабованих сховищ. Саме це створює передумови для побудови стійкої, модульної та науково обґрунтованої архітектури системи у наступних розділах.

2.2 Діаграма класів та кооперації інформаційної системи

Логічна структура програмного емулятора інтелектуальної системи розпізнавання об'єктів для складських підприємств реалізується у вигляді ієрархії класів, яка забезпечує слабке зв'язування між підсистемами, високу когерентність відповідальностей і можливість розширення функціональності без руйнування наявного коду. На рис. 2.2 подано діаграму класів ядра емулятора, що відображає розподіл обов'язків між компонентами сцени, конфігурацією сценарію, візуальним конвеєром, модулем телеметрії, трекером об'єктів, емулятором автопілота та підсистемою журналювання.

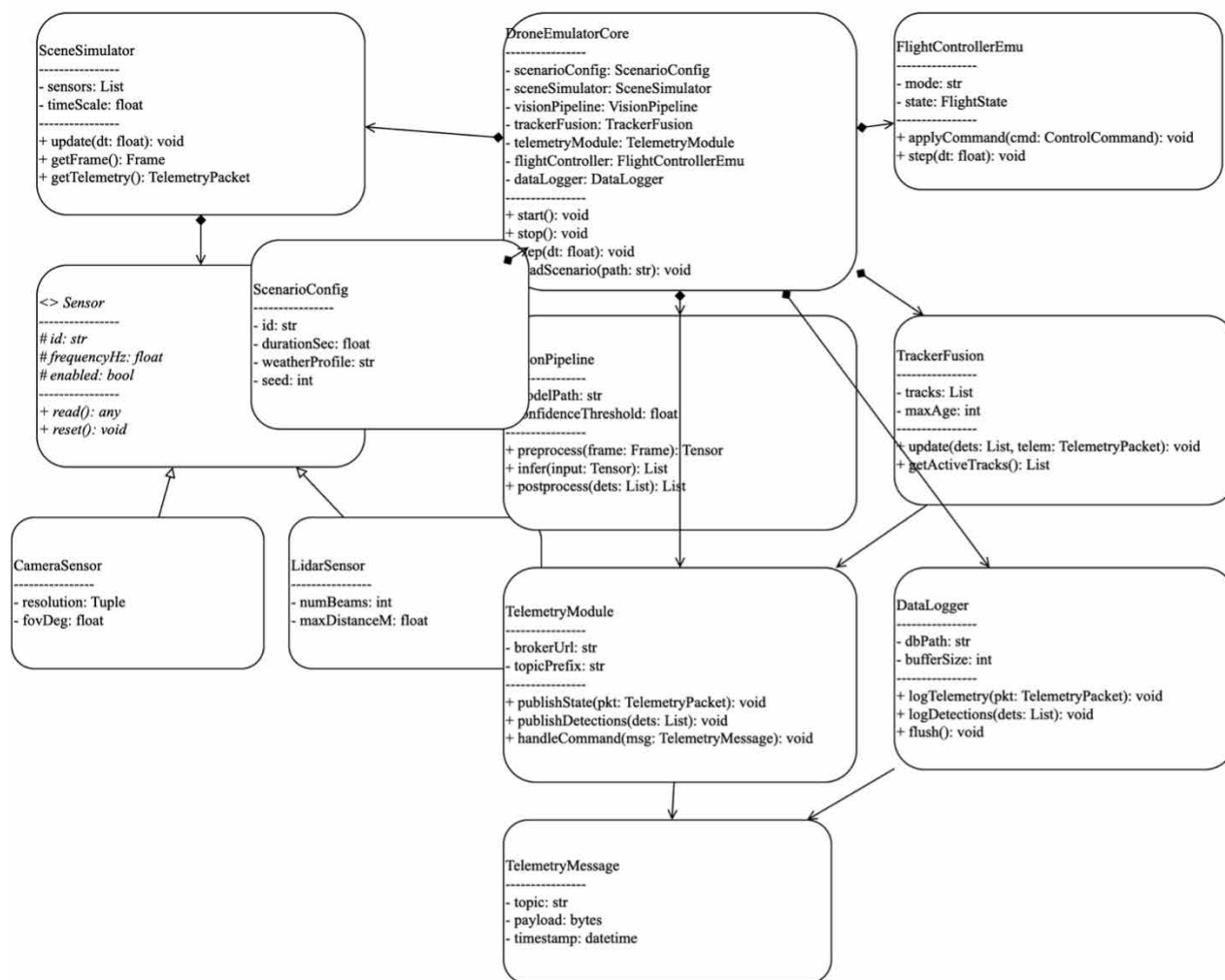


Рис. 2.2. UML-діаграма класів ядра емулятора інтелектуальної системи розпізнавання об'єктів

Клас `DroneEmulatorCore` відіграє роль фасаду та координатора, інкапсулюючи роботу з конфігурацією сценарію, симулятором сцени, візуальним конвеєром, модулем телеметрії, контролером польоту та журналом даних. Таке рішення дозволяє ізолювати зовнішні інтерфейси від внутрішньої складності системи й спрощує інтеграцію з графічним інтерфейсом користувача або MLOps-сервісами. Підсистема симуляції (`SceneSimulator`, абстрактний клас `Sensor` та його нащадки `CameraSensor` і `LidarSensor`) моделює фізичні сенсори, але представляє їх через уніфікований інтерфейс читання даних, що відповідає «нормалізації» на рівні об'єктно-орієнтованого проектування: різні джерела даних зводяться до узгодженого DOM-подання, яке далі сприймається візуальним конвеєром.

Важливою частиною архітектури є клас `VisionPipeline`, який капсулює етапи препроцесингу, інференсу й постпроцесингу детекцій. Наявність параметризованих атрибутів (модель, порог довіри, тощо) дає змогу легко переключати моделі YOLO/SSD, змінювати режими роботи (наприклад, лише детекція або детекція з трекінгом) без модифікації решти системи. Клас `TrackerFusion` реалізує логіку об'єднання детекцій у стійкі треки, використовуючи телеметричні пакети як додатковий контекст, що дає змогу поєднати зорову та навігаційну інформацію. `TelemetryModule`, `TelemetryMessage` та `DataLogger` утворюють узгоджену підсистему обміну та зберігання даних: перший виступає адаптером до зовнішніх транспортів (MQTT/UDP/файли), другий описує уніфікований формат повідомлення, третій забезпечує буферизацію й транзакційний запис у сховище. Така декомпозиція зменшує взаємну залежність компонентів та створює основу для масштабування (додавання нових каналів, форматів логів, типів повідомлень).

Для аналізу поведінки системи розглянуто кілька ключових кооперацій між об'єктами. На рис. 2.3 подано діаграму кооперації (послідовності) конвеєра комп'ютерного зору, де компоненти `FrameBuffer: StreamAdapter`, `Detector: CVPipeline`, `Tracker: ObjectTracker`, `Classifier: MLModel` та `ObjectRepository: StateStore` реалізують базовий цикл сприйняття.

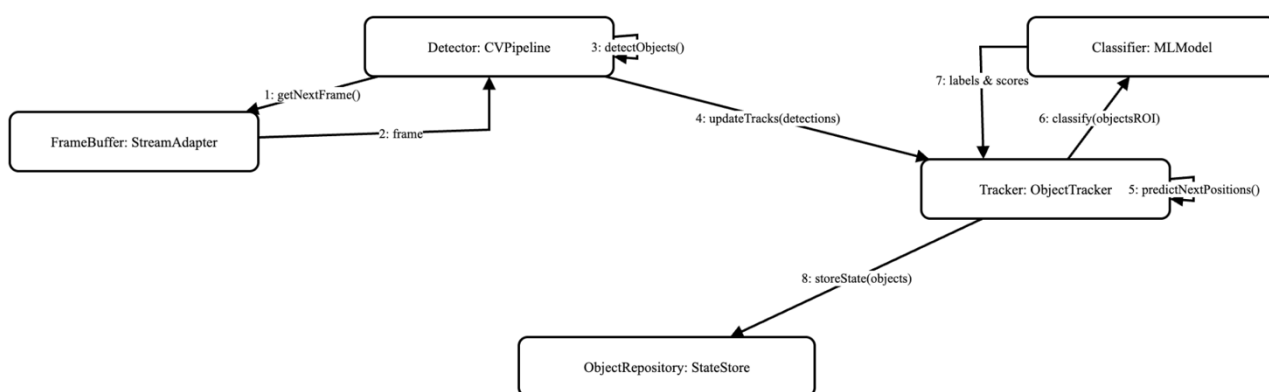


Рис. 2.3. Діаграма кооперації конвеєра комп'ютерного зору в емуляторі

У цьому сценарії `FrameBuffer` виступає абстракцією джерела кадрів (як реального відеопотоку, так і синтетичної сцени), тоді як `Detector` виконує інференс, `Tracker` стабілізує траєкторії та прогнозує позиції, `Classifier` за потреби

уточнює клас об'єкта, а `ObjectRepository` зберігає актуальний стан для подальшого використання системою планування місії. Така кооперація відображає принцип «розділення відповідальностей»: кожен клас виконує вузьку, чітко окреслену функцію, що спрощує тестування та подальше вдосконалення алгоритмів.

На рис. 2.4 наведено кооперацію компонентів каналу передачі відео між БПЛА та наземною станцією.

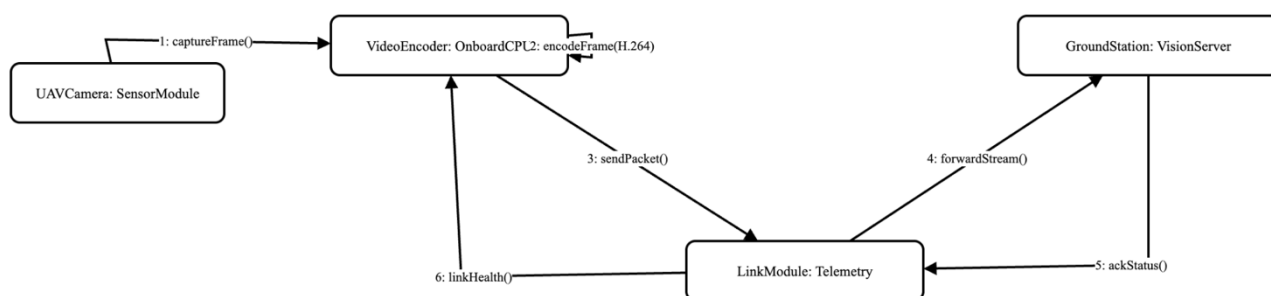


Рис. 2.4. Діаграма кооперації модуля відеопотоку між бортовою камерою та наземною станцією

У цьому сценарії клас `UAVCamera: SensorModule` захоплює кадри, `VideoEncoder: OnboardCPU2` виконує кодування (наприклад, H.264), `LinkModule: Telemetry` транспортує пакети, а `GroundStation: VisionServer` приймає й декодує потік, забезпечуючи оператору можливість візуального контролю та відлагодження роботи інтелектуальних модулів. Кооперація показує, що система спроектована таким чином, аби модулі кодування й транспортування були ізольовані від комп'ютерного зору, що дозволяє незалежно оптимізувати мережевий стек і алгоритми аналізу зображень.

На рис. 2.5 подано діаграму кооперації підсистеми прийняття рішень і управління польотом, в якій `ObjectStateService: DataAPI`, `MissionPlanner: DecisionEngine`, `Navigator: FlightControl` та `Autopilot: UAVController` реалізують контур «сприйняття – планування – виконання».

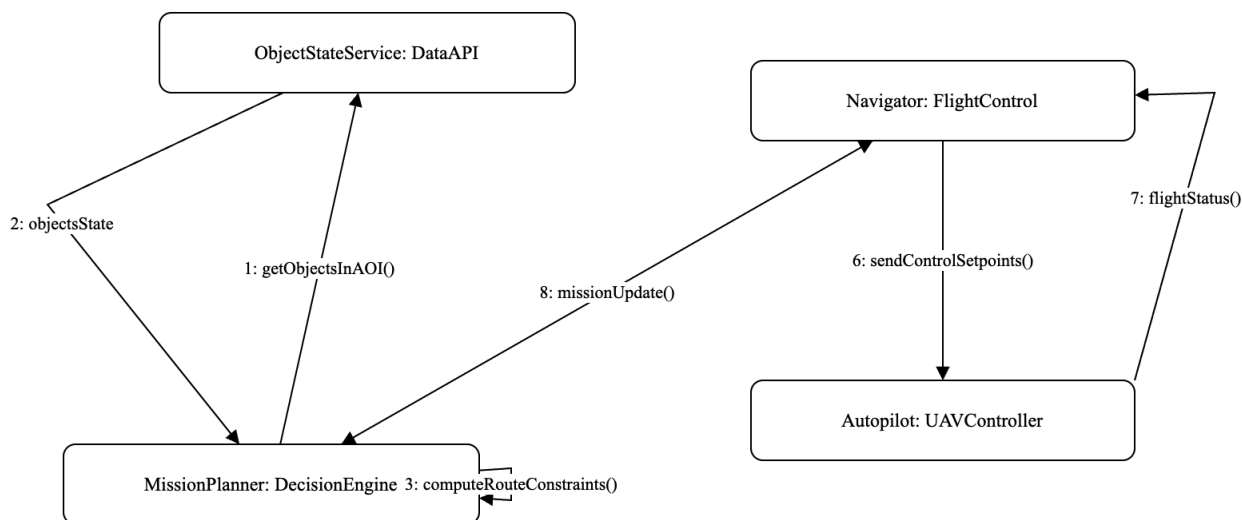


Рис. 2.5. Діаграма кооперації підсистеми планування місії та управління польотом

У цьому сценарії сервіс стану об'єктів надає агреговані дані про виявлені цілі та перешкоди, рушій прийняття рішень обчислює обмеження маршруту й оновлює план місії, навігатор транслює керуючі точки у команди низького рівня, а блок автопілота реалізує їх на рівні двигунів та поверхонь керування. Така багат шарова архітектура дозволяє відокремити стратегічну логіку від тактичного управління, що є критично важливим для перевірки безпеки алгоритмів та їх подальшої сертифікації.

Для систематизації структури було сформовано узагальнену таблицю основних класів, наведену у табл. 2.2, де вказано їх приналежність до певної підсистеми та базову роль у загальній архітектурі.

Таблиця 2.2

Основні класи та їх приналежність до підсистем інтелектуального емулятора

| Клас / група класів | Підсистема | Узагальнена роль |
|---|----------------------|--|
| DroneEmulatorCore, ScenarioConfig, SceneSimulator | Ядро емулятора | Координація сценарію, керування життєвим циклом симуляції, ін'єкція залежностей між модулями |
| Sensor, CameraSensor, LidarSensor | Моделювання сенсорів | Формування уніфікованих потоків даних для подальшої обробки візуальним конвесором |

Продовження таблиці 2.2

| | | |
|---|---------------------------|--|
| VisionPipeline, ObjectTracker, Classifier, FrameBuffer | Комп'ютерний зір | Препроцесинг, інференс, трекінг і класифікація об'єктів, підтримка реального часу |
| TelemetryModule, TelemetryMessage, LinkModule | Комунікації та телеметрія | Транспортування відео й телеметричних пакетів між бортом та наземною інфраструктурою |
| DataLogger, ObjectRepository, ObjectStateService | Зберігання та аналітика | Надійне логування, зберігання стану об'єктів, підготовка даних для MLOps і планувальника |
| MissionPlanner, Navigator, Autopilot, FlightControllerEmu | Управління польотом | Планування маршруту, трансляція рішень у керуючі команди, емуляване виконання маневрів |

Результуючи, можна зазначити, що розроблена діаграма класів і відповідні діаграми кооперацій відображають цілісну архітектуру програмного емулятора, у якій логіка сприйняття, передачі даних, зберігання стану та управління польотом рознесені по спеціалізованих модулях із чітко визначеними інтерфейсами. Такий підхід забезпечує «нормалізацію» на рівні об'єктно-орієнтованої моделі: кожен клас має одиничну відповідальність, асоціації між класами мінімізують надлишкові залежності, а кооперації формують зрозумілі сценарії виконання, які легко верифікувати й розширювати. У підсумку це створює надійну основу для реалізації інтелектуальної системи розпізнавання об'єктів та її інтеграції як у симуляційні, так і в реальні контури управління БПЛА, забезпечуючи масштабованість, тестованість і можливість подальшого наукового вдосконалення алгоритмів.

2.3 Діаграма компонентів інформаційної системи

Архітектура інтелектуальної системи розпізнавання об'єктів для складських підприємств базується на модульному підході, що забезпечує

незалежність, масштабованість і можливість подальшої еволюції компонентів без порушення цілісності системи. На рис. 2.3 подано діаграму компонентів, яка відображає взаємодію основних підсистем: модуля сенсорної емуляції, підсистеми комп'ютерного зору, каналу телеметрії та обміну, сховища даних, контролера польоту, підсистеми MLOps і графічної наземної станції.

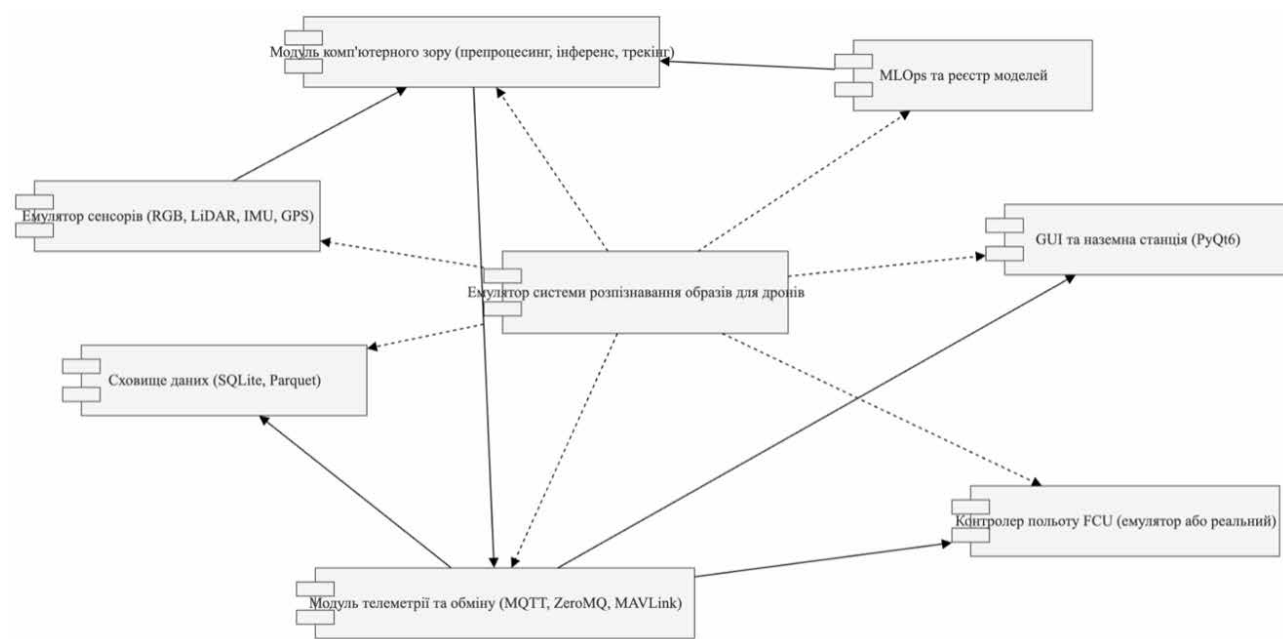


Рис. 2.3. UML-діаграма компонентів інтелектуальної системи розпізнавання об'єктів для БПЛА.

Щоб систематизувати роль програмних компонентів, доцільним є узагальнення їх функціонального призначення у табл. 2.3, де наведено основні компоненти, їх інтерфейси та логіку взаємодії з ядром емулятора.

Таблиця 2.3

Основні компоненти архітектури системи та їх функціональні ролі

| Компонент | Призначення | Основні інтерфейси |
|--|---|--|
| Емулятор сенсорів (RGB, LiDAR, IMU, GPS) | Генерація синтетичних потоків даних для тестування моделей комп'ютерного зору та алгоритмів керування | SensorAPI, FrameStream, TelemetryFeed |
| Модуль комп'ютерного зору | Препроцесинг, інференс, трекінг об'єктів; формування структурованих результатів детекції | VisionAPI, ModelRunner, TrackingModule |

Продовження таблиці 2.3

| | | |
|---|---|--|
| Модуль телеметрії та обміну (MQTT / ZeroMQ / MAVLink) | Передача відео, детекцій, команд та телеметричних пакетів між компонентами системи | TelemetryBus, CommandLink, StreamPublisher |
| Контролер польоту (емулятор або реальний FCU) | Виконання навігаційних команд, стабілізація польоту, взаємодія з автопілотом | MAVLinkAPI, ControlSignals |
| Сховище даних (SQLite, Parquet) | Збереження телеметрії, логу інференсу, фреймів, метаданих та результатів аналізу | StorageDriver, QueryInterface |
| MLOps та реєстр моделей | Керування версіями моделей, оновлення конвеєра інференсу, адаптація параметрів для симуляцій | ModelRegistry, DeploymentAPI |
| GUI та наземна станція (PyQt6) | Моніторинг стану системи, візуалізація детекцій, перегляд телеметрії та журналів | GroundStationAPI, VisualizationModule |
| Емулятор системи розпізнавання образів для дрона (ядро) | Координація роботи всіх модулів, синхронізація потоків даних, оркестрація сценаріїв симуляції | CoreAPI, IntegrationBus |

Узагальнюючи, компонентна архітектура системи базується на принципах модульності, інкапсуляції й слабкої зв'язності, що дає змогу незалежно масштабувати окремі підсистеми - від телеметрії й зберігання даних до візуального конвеєра та модулів MLOps. Така структура забезпечує можливість тестування моделей у різних сценаріях без потреби змінювати апаратний контур, а також створює повноцінну платформу для досліджень комп'ютерного зору в контексті автономної навігації БПЛА. Компоненти взаємодіють через стабільні інтерфейси, що спрощує інтеграцію зовнішніх модулів (реальних сенсорів, інших емуляторів, хмарних сервісів), підвищує надійність системи та дозволяє реалізувати MLOps-підхід у циклі “симуляція → інференс → оцінювання → оновлення моделі”.

2.4 Діаграма пакетів

Архітектурна модель програмного комплексу передбачає чітке логічне групування функціональності в окремі пакети, що забезпечує модульність, повторне використання коду та мінімізацію зв'язності між підсистемами. На рис. 2.4 наведено UML-діаграму пакетів, яка відображає структуру ядра емулятора, модулів комп'ютерного зору, телеметрії, зберігання даних, керування конфігураціями та інтеграції з MLOps-компонентами.

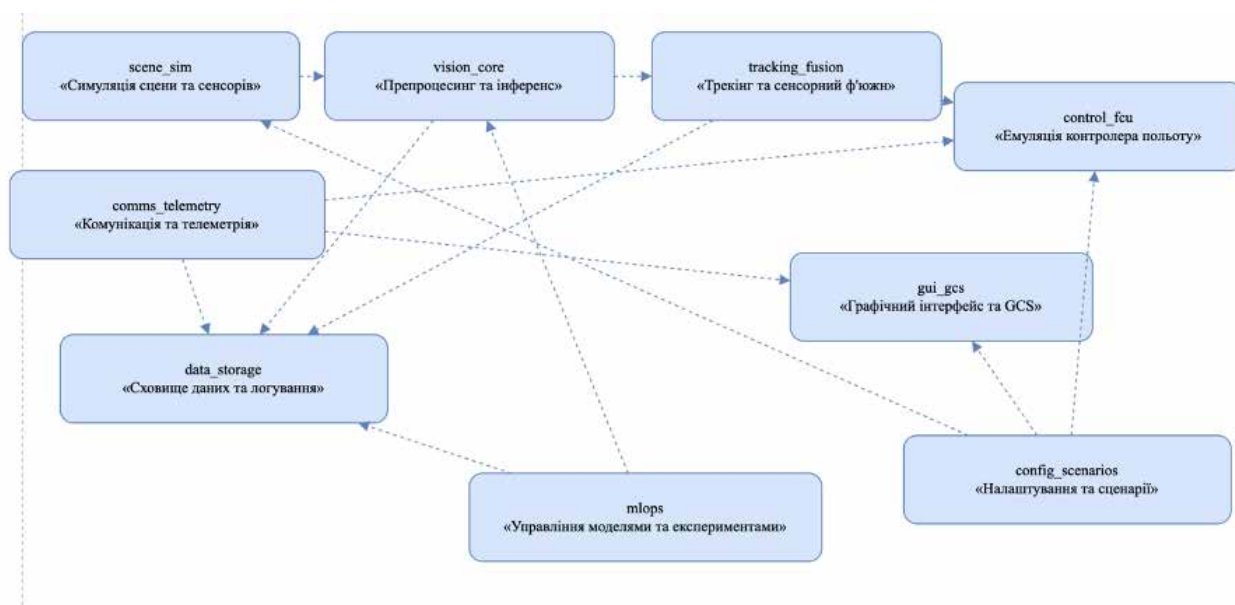


Рис. 2.4. Діаграма пакетів інтелектуальної системи розпізнавання об'єктів для БПЛА.

Для узгодженого представлення структури системи у табл. 2.4 наведено призначення кожного пакета та спектр його відповідальностей у межах всієї програмної архітектури.

Таблиця 2.4

Призначення пакетів інтелектуальної системи та їх відповідальності

| Пакет | Призначення | Основні зони відповідальності |
|-----------|---|---|
| scene_sim | Формування віртуального середовища та емуляція сенсорної інформації | Генерація фреймів, обробка шумів, моделювання LiDAR/IMU |

Продовження таблиці 2.4

| | | |
|------------------|---|--|
| vision_core | Обробка даних комп'ютерного зору та інференс моделей | Препроцесинг, inference, детекції, формування ROI |
| tracking_fusion | Сегментація, трекінг і сенсорний ф'юзж | Об'єднання даних з різних сенсорів, прогноз траєкторій |
| control_fcu | Емуляція контролера польоту та реакція на команди | Взаємодія з MAVLink API, стабілізація польоту |
| comms_telemetry | Комунікація між модулями та потокова передача даних | MQTT/ZeroMQ/MAVLink, стрімінг телеметрії та відео |
| data_storage | Централізоване зберігання телеметрії та логів інференсу | SQLite/Parquet драйвери, логування, буферизація |
| gui_gcs | Графічний інтерфейс наземної станції | Візуалізація, моніторинг, управління місією |
| mlops | Керування моделями та експериментами | Реєстр моделей, версіонування, оновлення конвеєра |
| config_scenarios | Налаштування параметрів симуляції, сценарії польоту | Конфігурації середовища, профілі експериментів |

У результаті отримано структуровану, масштабовану та формально коректну архітектурну модель, у якій кожен пакет виконує чітко визначену роль та взаємодіє з іншими через стабільні інтерфейси. Така організація суттєво спрощує інтеграцію нових алгоритмів комп'ютерного зору, підключення додаткових сенсорів, зміну конфігурацій польотних сценаріїв та модернізацію системи у межах дослідницького циклу MLOps. Завдяки логічній декомпозиції через пакети забезпечено узгодженість архітектури, повторне використання модулів та можливість незалежної розробки складових частин системи.

2.5 Висновки до другого розділу

У другому розділі було здійснено всебічне проектування архітектури інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими запасами, що охоплює побудову логічної моделі даних,

формування діаграм класів і кооперацій, а також структурування компонентів та пакетів відповідно до вимог предметної області та сучасних підходів системного моделювання. Логічна модель даних (рис. 2.1) відобразила чітку семантичну структуру складських процесів, у якій кожна сутність — товар, SKU, партія, відеофрейм, результат детекції, зона зберігання чи запис інвентаризації — має уніфіковані атрибути, первинні та зовнішні ключі, нормалізовані зв'язки та однозначну інтерпретацію у контексті задач детекції, трекінгу та синхронізації з WMS. Дотримання принципів третьої нормальної форми забезпечило мінімізацію дублювання даних, підвищення узгодженості інформаційної моделі та коректне відображення взаємодії між потоками відеоданих, складськими операціями та результатами інтелектуального аналізу.

Діаграми класів і кооперацій (рис. 2.2–2.5) дали змогу формалізувати структурну й поведінкову логіку ключових компонентів системи, зокрема модулів комп'ютерного зору, підсистеми трекінгу, аналітики заповненості стелажів, модулів оброблення відеопотоків, підсистеми синхронізації з WMS та MLOps-сервісів. Побудовані моделі продемонстрували, як об'єкти системи взаємодіють у межах циклу «приймання відеоданих → інференс → підрахунок → трекінг → аналітика → оновлення залишків WMS → логування», відображаючи алгоритмічну послідовність, розподіл відповідальності між класами та взаємозв'язки між підсистемами. Таке моделювання забезпечило прозорість внутрішніх механізмів системи та покращило прогнозованість її поведінки при роботі з великими обсягами складських даних.

Діаграма компонентів та діаграма пакетів (рис. 2.6–2.7) сформували цілісне уявлення про модульно-орієнтовану архітектуру майбутнього рішення, у якій функціональність розподілена між незалежними, але взаємопов'язаними підсистемами — обробленням відеопотоків, інтелектуальною детекцією, трекінгом, аналітикою складських зон, інтеграцією з WMS, зберіганням даних, Edge-компонентами та MLOps. Кожен компонент має чітко визначений інтерфейс взаємодії, що мінімізує зв'язність, спрощує тестування та розширення

системи, а пакетна структура впорядковує логіку засобами low coupling / high cohesion, закладаючи основу для масштабованої та гнучкої архітектури.

Другий розділ сформував фундаментальну архітектуру інтелектуальної системи, яка визначає подальшу реалізацію, тестування та експериментальну валідацію програмного комплексу. Результати моделювання забезпечують формально обґрунтовану, послідовну та логічно цілісну основу для побудови системи з високими вимогами до точності інференсу, стабільності роботи, відмовостійкості та можливості масштабування у реальних умовах функціонування складських комплексів. Побудована архітектура створює необхідні передумови для впровадження інтелектуальної автоматизації управління запасами, підвищення ефективності логістичних процесів та мінімізації людського фактору в операціях великої інтенсивності.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технологій та інструментальних засобів реалізації системи

Проєктування інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими підприємствами вимагає ретельного добору технологій, які одночасно забезпечують високу продуктивність, надійність, відмовостійкість та гнучкість під час роботи з потоковими сенсорними даними. Враховуючи специфіку задач — багатопоточний прийом телеметрії, оброблення відеопотоку в реальному часі, виконання інференсу моделей глибокого навчання, синхронізація трекерів та модулів прийняття рішень - було обрано сучасний стек, який вже продемонстрував відповідність промисловим вимогам у сфері UAV, робототехніки та комп'ютерного зору.

Основою реалізації є мова Python, що забезпечує швидку розробку, наявність численних бібліотек для CV і ML та тісну інтеграцію з апаратними сенсорами. Бібліотеки PyTorch та ONNX Runtime дають змогу виконувати інференс оптимізованих моделей (YOLOv8s/SSD-MobileNet) на CPU або GPU, мінімізуючи затримки. Інструменти симуляції сенсорів реалізовано із використанням NumPy, OpenCV та custom scene generator, що дозволяє відтворювати повністю контрольовані сценарії тестування. Обмін телеметрією побудовано на протоколах MQTT, ZeroMQ та MAVLink, які залишаються стандартом у сфері безпілотної авіації та гарантують стабільність і низьку латентність. Для логування та зберігання результатів детекцій використано SQLite, а для великих масивів сенсорних даних — формат Parquet, що оптимізує роботу із часо-серійними вибірками. Графічний інтерфейс реалізується засобами PyQt6, що дозволяє створити повноцінну наземну станцію управління та візуалізації.

Підсистема управління моделями передбачає використання MLflow або аналога для відстеження версій, параметрів та експериментів, що дає змогу інтегрувати процеси MLOps у життєвий цикл розробки. Вибрані технології формують узгоджену архітектуру, у якій модулі можуть розширюватися, оптимізуватися та інтегруватися з реальними компонентами UAV без суттєвих змін у кодовій базі.

Таблиця 3.1

Основні технології та їх функціональне призначення

| Технологія / інструмент | Призначення в системі |
|-------------------------|--|
| Python 3.x | Базова мова реалізації всіх модулів симуляції, телеметрії, CV та логування |
| PyTorch / ONNX Runtime | Виконання інференсу моделей глибокого навчання; підтримка CPU/GPU |
| OpenCV, NumPy | Препроцесінг відеофреймів, генерація синтетичних даних, робота із сенсорними матрицями |
| MQTT, ZeroMQ, MAVLink | Потоковий обмін телеметрією та службовими повідомленнями між моделями та емуляторами |
| SQLite, Parquet | Постійне зберігання місій, фреймів, треків та телеметрії; ефективна робота з великими наборами даних |
| PyQt6 | Реалізація графічної наземної станції, візуалізація польоту та результатів CV |
| MLflow / Model Registry | Управління версіями моделей, параметрами інференсу, експериментами |
| FastAPI (опційно) | REST-взаємодія з сервісами, централізований доступ до сховищ і станів |
| Docker | Контейнеризація компонентів, забезпечення переносності та стабільності середовища |

Обраний стек технологій забезпечує баланс між продуктивністю інференсу, гнучкістю розробки та можливістю масштабування системи під різні сценарії використання — від лабораторних емуляцій до реальних польотних місій. Використання Python та його спеціалізованих бібліотек гарантує інтеграцію усіх модулів у єдину архітектуру, тоді як стандартизовані протоколи телеметрії та оптимізовані сховища підвищують надійність і відтворюваність

експериментів. Таким чином, вибір технологій є обґрунтованим і повністю відповідає вимогам до сучасних автономних систем комп'ютерного зору для безпілотних літальних апаратів.

3.2 Архітектура системи, проєктування функціоналу результатів дослідження

Архітектура інтелектуальної системи розпізнавання об'єктів формувалася як багаторівнева модульна структура, що поєднує емуляційне ядро, модулі сенсорного моделювання, підсистеми телеметрії, модулі комп'ютерного зору, елементи трекінгу та сенсорного злиття, а також засоби зберігання, логування та управління моделями. Загальна компоновальна схема ядра емулятора наведена на рис. 3.1, де відображено взаємодію модулів симуляції сцен і сенсорів, емулятора контролера польоту, джерел синтетичного відеопотоку, телеметричного каналу, модулів інференсу та трекінгу, а також підсистеми MLOps та сховища даних.

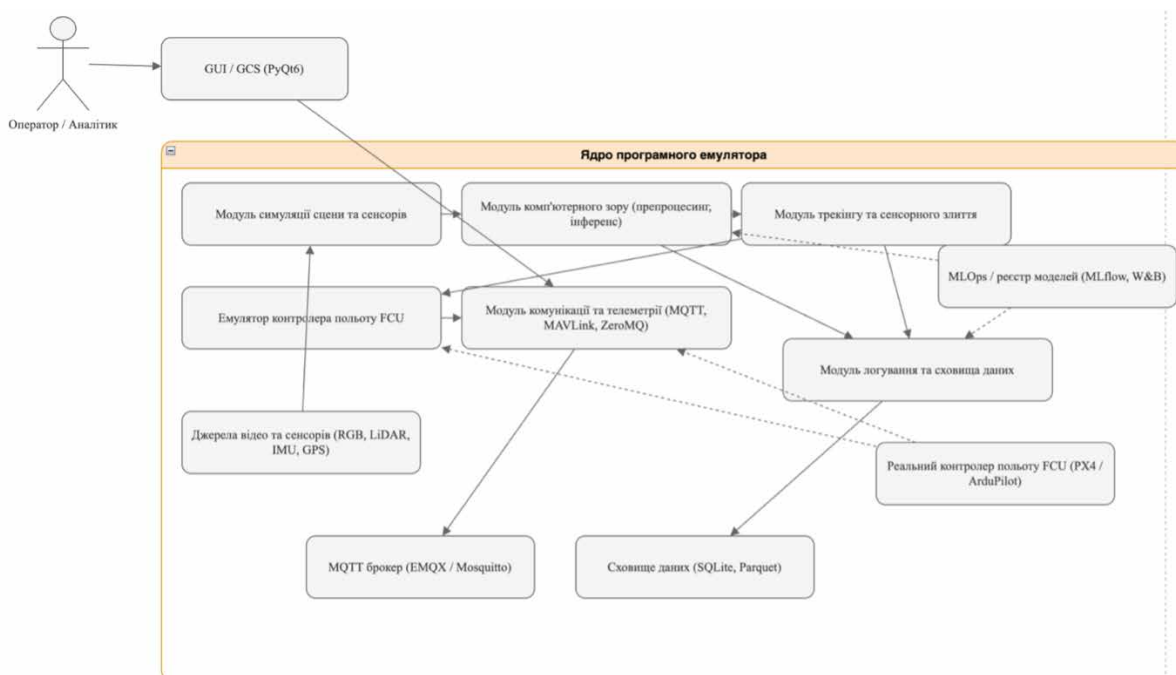


Рис. 3.1. Архітектура ядра програмного емулятора системи розпізнавання об'єктів

Інтеграція інтелектуальних модулів з каналами управління та телеметрії ВПЛА зображена на рис. 3.2, де відображено взаємодію автономного контролера польоту, інтерфейсу сенсорних модулів, станції наземного управління, серверів комп'ютерного зору та сховища місій і телеметрії. Ця схема дозволяє простежити наскрізний ланцюг даних — від формування сирих сенсорних кадрів до прийняття рішень, оновлення маршруту та передавання керуючих команд.

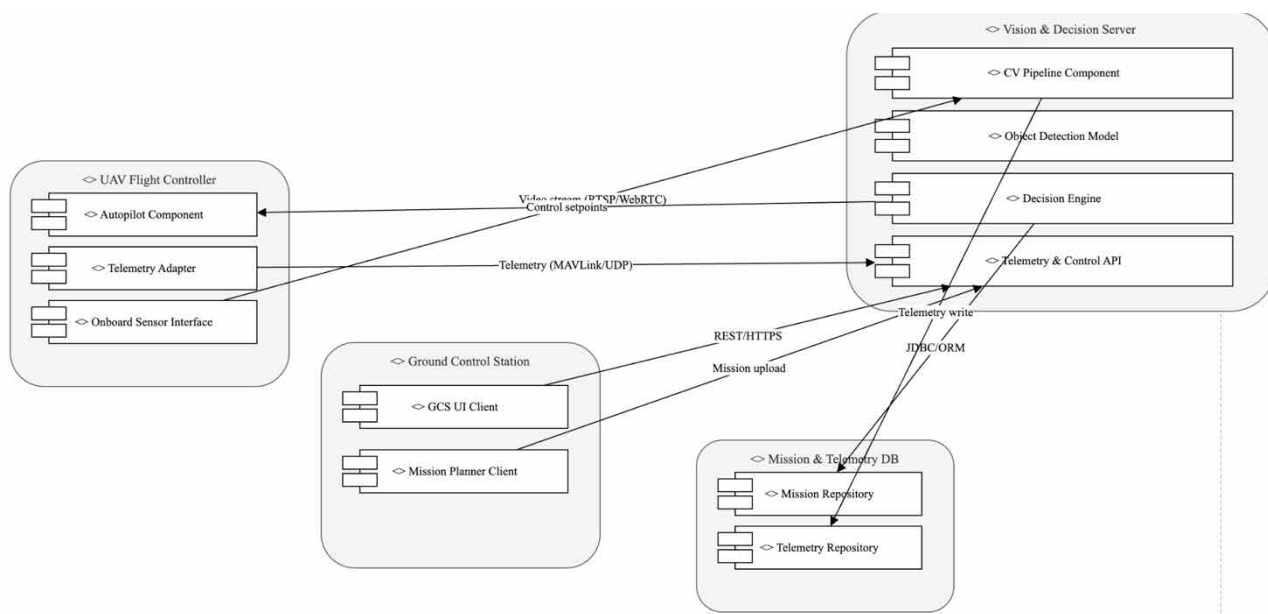


Рис. 3.2. Інтеграція модулів комп'ютерного зору, телеметрії та управління БПЛА

Структура проєктування була побудована відповідно до результатів моделювання предметної області та сформованих вимог до функціональності інтелектуальної системи управління складськими запасами. Основним принципом стала модульність та незалежність компонентів: ядро підсистеми відеоаналітики приймає потоки з камер та одразу передає їх до CV-конвеєра; модуль інтеграції з WMS забезпечує асинхронну взаємодію з інформаційною системою складу; трекінговий модуль синхронізує результати детекцій із зональною структурою простору та динамікою переміщення товарів; підсистема логування накопичує відеодані, метрики інференсу, журнали операцій і результати аналітики для подальшого аналізу, контролю якості та MLOps-процедур. Такий поділ дозволяє спростувати масштабування, заміну окремих

компонентів та адаптацію системи під різні сценарії складської роботи, інтенсивність потоків та типи продукції.

Архітектура забезпечує чітке розмежування відповідальностей: підсистема приймання даних відповідає за стабілізацію зображення та коректне формування відеопотоку; модулі комп'ютерного зору — за точність та швидкість розпізнавання товарних одиниць; трекінговий модуль — за узгодження просторово-часових ознак та формування траєкторій переміщення товарів у зонах комплектування, сортування та зберігання; модуль взаємодії з WMS — за автоматизоване оновлення залишків, фіксацію інвентаризаційних відхилень та формування запитів до ERP. Водночас реалізовано наскрізну підтримку експериментального циклу: кожен етап — від оброблення відеопотоку до аналізу заповненості стелажів — повністю логований, що дає змогу виконувати стратегічну оптимізацію моделей, оцінювати стабільність детекцій, аналізувати випадки помилкових класифікацій і досліджувати вплив умов освітлення, щільності товарів та динаміки операцій на точність CV-алгоритмів.

Результати проектування підтвердили доцільність комплексного підходу, за якого модулі машинного навчання функціонують у тісній зв'язці з підсистемами аналітики, WMS-інтеграції та керування складськими процесами, утворюючи єдину інтегровану архітектуру. Такий підхід дозволяє не лише моделювати поведінку системи у складних логістичних сценаріях, а й забезпечує можливість роботи з високодинамічними потоками товарів, зміною розташування, появою нових SKU та аномалій у зонах зберігання. Завдяки модульності архітектури забезпечено підтримку різних моделей детекції (YOLO, EfficientDet, Transformer-моделі), можливість підключення додаткових сенсорів (Depth-камер, стереосистем, сканерів ваги чи об'єму), а також інтеграцію з MLOps-середовищами для виконання версіонування, автоматизованої перевірки моделей, відтворення експериментів та порівняння продуктивності різних конфігурацій. Це формує основу для побудови стабільної, масштабованої та адаптивної інтелектуальної системи, здатної ефективно функціонувати у реальних умовах сучасних складських комплексів.

3.3 Інформаційна база системи

Інформаційна база інтелектуальної системи розпізнавання об'єктів для автоматизації управління складами формується як багаторівневий комплекс структурованих і неструктурованих даних, що забезпечують повноцінну роботу модулів комп'ютерного зору, телеметрії, симуляції та аналітичних компонентів. Вона охоплює вхідні сенсорні потоки (RGB, LiDAR, IMU, GPS), параметри польотних місій, результати інференсу моделей, треки об'єктів, метрики продуктивності та аналітичні показники, необхідні для адаптивного управління і навчання моделей. Кластеризаційний аналіз, що використовується для побудови структурної моделі об'єктного середовища, відіграє ключову роль у формуванні інформаційної бази, оскільки дозволяє виокремлювати закономірності у поведінці об'єктів, шумовій складовій даних і параметрах траєкторій. На рис. 3.6 подано результати кластеризації синтетичних об'єктів, які демонструють формування чітко розмежованих груп за двома ключовими ознаками, що є основою для подальшого трекінгу та фільтрації.



Рис. 3.6. Результати кластеризації об'єктів у просторі ознак

Подальший елемент інформаційної бази — визначення оптимальної кількості кластерів, що здійснюється методом «лікоть». На рис. 3.7 наведено графік залежності WCSS від параметра k , який дозволяє мінімізувати надлишковість структуризації та одночасно зберегти семантичну цілісність локальних груп об'єктів. Така оптимізація дозволяє адаптивному трекеру коректно підбирати модель шуму та параметри предикції для реальних польотів.

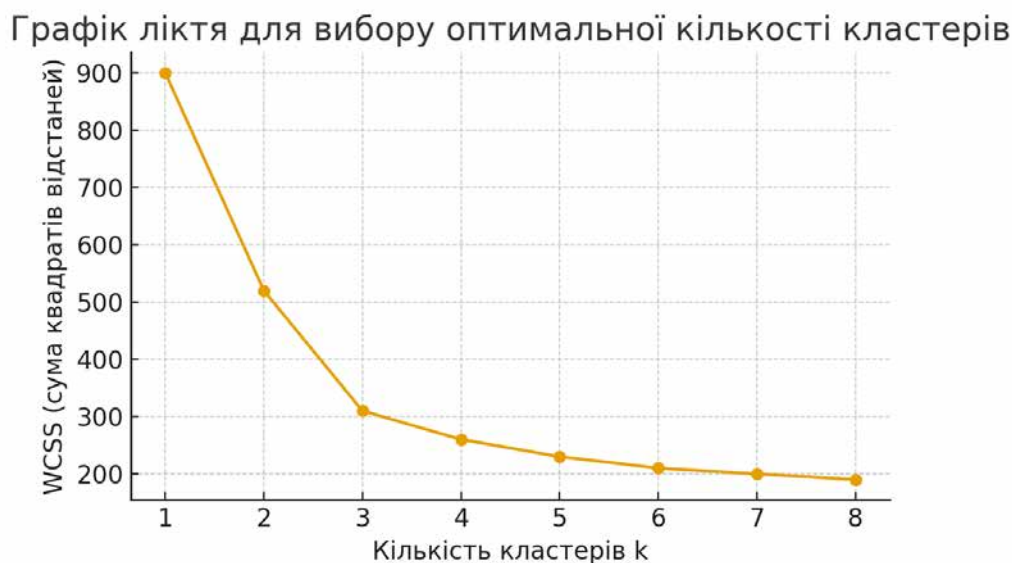


Рис. 3.7. Графік методу «лікоть» для визначення оптимального числа кластерів

У межах інформаційної бази формується модуль порівняльного аналізу моделей розпізнавання, що дозволяє системі приймати рішення щодо вибору оптимального CV-модуля для конкретних умов польоту, рівня шуму, ресурсу обчислень та пропускної здатності каналу телеметрії. На рис. 3.8 наведено радіальну діаграму порівняння моделей за шістьма критичними параметрами (точність, стійкість до шуму, швидкодія, енергоспоживання, стабільність трекінгу тощо), що ілюструє принцип прийняття інтегрованого рішення на основі багатовимірної оптимізації.

Порівняння характеристик моделей розпізнавання об'єктів

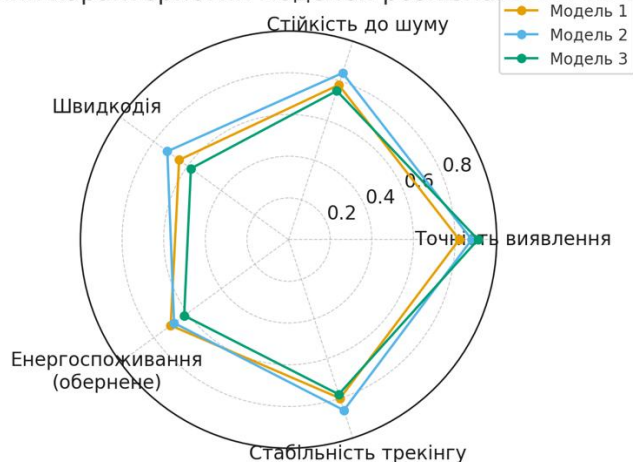


Рис. 3.8. Порівняльна радіальна діаграма характеристик моделей розпізнавання

Інформаційна база системи має чітку багаторівневу організацію, що поєднує сенсорні дані, похідні ознаки, результати CV-модулів та багатовимірну аналітику.

Результуючий аналіз засвідчує, що інформаційна база системи не є простою сукупністю даних, а виступає інтегрованим інтелектуальним ресурсом, який забезпечує адаптивність CV-модуля, стабільність трекінгу, підвищену точність предикції траєкторій та стійкість роботи всієї системи в умовах реального часу. Інтеграція методів кластеризації, багатокритеріального порівняння моделей і структурованого зберігання телеметрії створює основу для самонавчальних алгоритмів, що і формує головну наукову новизну дослідження — можливість автоматичного конфігурування CV-модуля залежно від контексту польотного середовища та параметрів об'єктної сцени.

3.3 Висновки до третього розділу

У третьому розділі було обґрунтовано вибір технологій, інструментальних засобів та архітектурних рішень, необхідних для створення повнофункціональної інтелектуальної системи розпізнавання об'єктів у контексті автоматизованого управління складськими запасами. Показано, що

інтеграція Python-екосистеми з PyTorch, OpenCV та ONNX Runtime забезпечує оптимальний баланс між швидкістю інференсу, стійкістю моделей та можливістю їх апаратної оптимізації на edge-платформах, що є критично важливим у високонавантажених складських середовищах. Використання PyQt6, REST/MQTT-комунікацій, SQLite/Parquet-сховищ та API-інтеграції з WMS формує узгоджене середовище для побудови інтерфейсів, оброблення потоків відеоданих та реалізації двосторонньої взаємодії з інформаційними системами складу.

Архітектурний аналіз підтвердив доцільність модульної організації системи, де підсистеми комп'ютерного зору, трекінгу товарних одиниць, інтеграції з WMS, ситуаційної аналітики, логування та оброблення потоків даних функціонують як автономні компоненти з чітко визначеними інтерфейсами. Такий підхід забезпечує можливість гнучкого масштабування відповідно до площі складу, інтенсивності потоків, кількості камер і складності логістичних сценаріїв, водночас дозволяючи застосовувати MLOps-процеси для версіонування, тестування та оптимізації моделей.

Особливу увагу приділено формуванню інформаційної бази, яка включає сирі відеодані, детекції, трекінгові траєкторії, метрики продуктивності, багатовимірні характеристики SKU та результати кластеризаційного аналізу. Було продемонстровано, що застосування методів кластеризації при аналізі структурованих складських даних підвищує точність просторової локалізації товарних одиниць та дозволяє адаптивно оптимізувати параметри моделей залежно від щільності викладки, зміни конфігурацій стелажів, інтенсивності потоків і різноманіття номенклатури. Метод «лікоть» дав змогу визначити оптимальну кількість груп для стабільної роботи алгоритмів агрегування даних, а радіальна порівняльна діаграма моделей стала основою для раціонального вибору архітектури CV-модуля під конкретні умови експлуатації.

Узагальнюючи результати, можна стверджувати, що третій розділ сформував завершену технологічну платформу, на якій ґрунтується програмно-алгоритмічне ядро системи. Було визначено архітектурні принципи, структуру

інформаційної бази, інструментальний стек та інноваційні методи роботи з даними й моделями, що забезпечують стійкий інференс, адаптивність до змінних умов складу та можливість інтеграції з корпоративними інформаційними системами. Наукова новизна розділу полягає у впровадженні механізмів адаптивної аналітики, здатних автоматично коригувати параметри розпізнавання залежно від логістичного навантаження та реальної структури складського простору, а також у створенні уніфікованої інфраструктури для підтримки повного життєвого циклу моделей комп'ютерного зору.

Таким чином, результати третього розділу забезпечують технологічну та методологічну основу для подальшої реалізації інтелектуальної системи управління складськими запасами, закладаючи фундамент для впровадження високоточних алгоритмів комп'ютерного зору, автоматизованої аналітики та інтегрованого прийняття рішень у сучасних складських комплексах.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

Тестування програмного емулятора інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими запасами передбачає комплексну перевірку всіх функціональних підсистем із урахуванням особливостей потокового опрацювання відеоданих, поведінки товарних потоків, варіативності умов освітлення та динаміки логістичних операцій. Оскільки система включає модулі відеозахоплення, препроцесингу, ML-інференсу, трекінгу товарних одиниць, інтеграції з WMS, логування, сховищ даних та GUI-інтерфейс для взаємодії оператора зі складською інфраструктурою, план тестування повинен забезпечити верифікацію як кожного окремого компонента, так і їх узгодженої взаємодії у режимі реального часу. Основна увага приділяється стабільності оброблення потоків відео, точності детекції та трекінгу SKU, відповідності модулів прогнозування очікуваним характеристикам, відтворюваності результатів та коректності синхронізації між підсистемами розпізнавання, аналітики та WMS-інтеграції.

У таблиці 4.1 наведено структурований план тестування, що охоплює функціональні, продуктивні, часові, комунікаційні та стресові випробування ключових програмних компонентів системи. Передбачено перевірку коректності роботи камер та препроцесингу, точності детекції об'єктів у складському середовищі з різною щільністю викладки, стабільності трекінгу під час інтенсивного переміщення товарів, узгодження даних між етапами оброблення та відсутності розсинхронізації часових міток. Окрему увагу приділено тестуванню інтеграційних механізмів із WMS, перевірці реакції системи на

змінне навантаження, поведінці під час пікових потоків товарів та коректності збереження результатів у базах SQLite і Parquet, що забезпечує можливість подальшого аналітичного опрацювання та використання в MLOps-процесах.

Таблиця 4.1

План тестування програмних модулів системи

| Модуль / підсистема | Ціль тестування | Основні метрики | Метод оцінювання |
|---|--|--|--|
| Модуль відеозахоплення та препроцесингу | Перевірка коректності стабілізації, нормалізації та роботи камер у різних умовах | FPS; latency; зміна гістограми; рівень шуму | Порівняння до/після обробки; аналіз стабільності частоти кадрів |
| Модуль інференсу CV | Перевірка точності детекції SKU та швидкодії моделей | mAP@50; latency; FPS; recall | Тестування на контрольних наборах SKU; профілювання інференсу |
| Модуль трекінгу | Перевірка злиття детекцій, стійкості треків та траєкторної узгодженості | MOTA; MOTP; % втрати треків | Порівняння з еталонними траєкторіями; аналіз стабільності в динаміці |
| Модуль інтеграції з WMS | Перевірка оновлення залишків, інвентаризації та передачі подій | Затримка обміну; узгодженість даних; % коректних записів | Імітація операцій складу; аналіз WMS-логів |
| Підсистема логування та сховище | Перевірка відтворюваності, цілісності та форматів записів | Цілісність даних; коректність timestamp; хеш-контроль | Повторні прогони; контроль хешів; відповідність форматів |
| Інтерфейс GUI | Перевірка коректності відображення детекцій, треків, статусів та взаємодії | Час оновлення UI; відповідність візуалізації | Аналіз реакції UI на реальний відеопотік і оновлення даних |

Методика оцінювання результатів передбачає комбіноване застосування кількісних метрик та візуального аналізу поведінки системи під час роботи у різних складських сценаріях. Для ML-компонентів використовуються mAP@50,

FPS, latency, recall та стійкість до шуму, що дозволяє визначити оптимальні моделі детекції для умов високої щільності товарів, руху персоналу та наявності візуальних перешкод. Для трекінгу застосовуються показники MOTA та MOTP, які характеризують точність та плавність відстежування SKU у просторі складських зон. Для каналів передачі даних критичними є параметри затримки, пропускної здатності та узгодженості транзакцій із WMS, що дозволяє оцінити вплив навантаження на коректність оновлення складських записів.

Узагальнюючи результати планування, можна стверджувати, що наведена методика формує всебічну основу для оцінювання працездатності системи за ключовими характеристиками — точністю, продуктивністю, масштабованістю, стресостійкістю та відтворюваністю. Такий підхід гарантує, що отримані результати адекватно відображають реальні властивості програмних модулів, їхню взаємодію в умовах реального складу та можуть бути використані для подальшої оптимізації архітектури, налаштування моделей і загального вдосконалення інтелектуальної системи управління складськими запасами.

4.2 Тестування інтелектуальної системи розпізнавання об'єктів у середовищі програмного емулятора

Тестування інтелектуальної підсистеми здійснювалося у розробленому програмному емуляторі, який забезпечує контрольоване відтворення сценаріїв польоту, генерацію сенсорних даних та інтеграцію з модулем комп'ютерного зору у режимі реального часу. На рис. 4.1 наведено фрагмент інтерфейсу підсистеми візуального моніторингу, що демонструє роботу механізмів детекції та трекінгу у сцені Urban Search & Rescue. Система у реальному часі відображає результати розпізнавання (Person, Vehicle, Backpack), середнє завантаження обчислювальних ресурсів, затримку інференсу та актуальні телеметричні параметри, що дозволяє оцінити стабільність алгоритмів у динамічному середовищі (рис. 4.1).

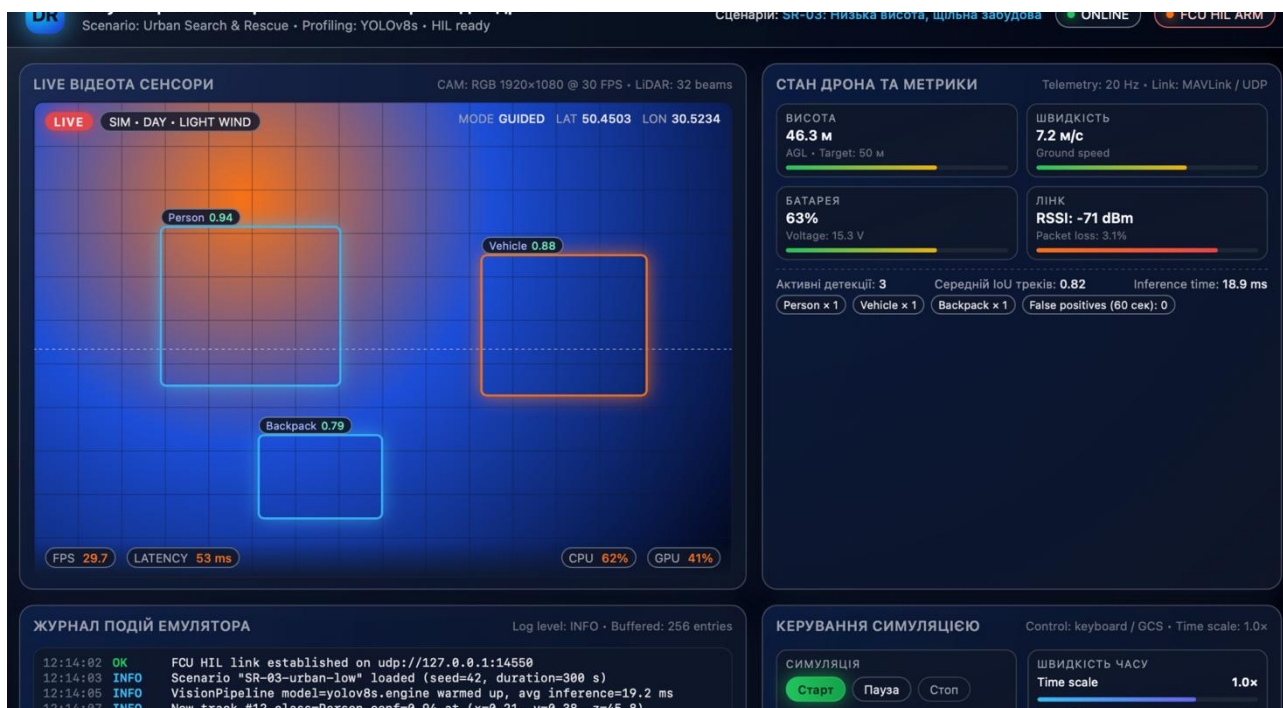


Рис. 4.1. Інтерфейс екрана «LIVE» з детекцією, телеметрією та оцінкою затримок

На рис. 4.2 показано інтегрований аналітичний екран продуктивності, який використовується для післяекспериментальної оцінки точності моделей, аналізу латентності, стабільності FPS, динаміки хибних спрацьовувань та формування матриці змішування для трьох класів (Person, Vehicle, Backpack). Використання такого панельного підходу забезпечує цілісне бачення продуктивності моделі в реальних умовах, включно з оцінкою P50/P95 латентності, середнім FPS, статистикою за сценаріями та характеристиками пропускної здатності логів (рис. 4.3).

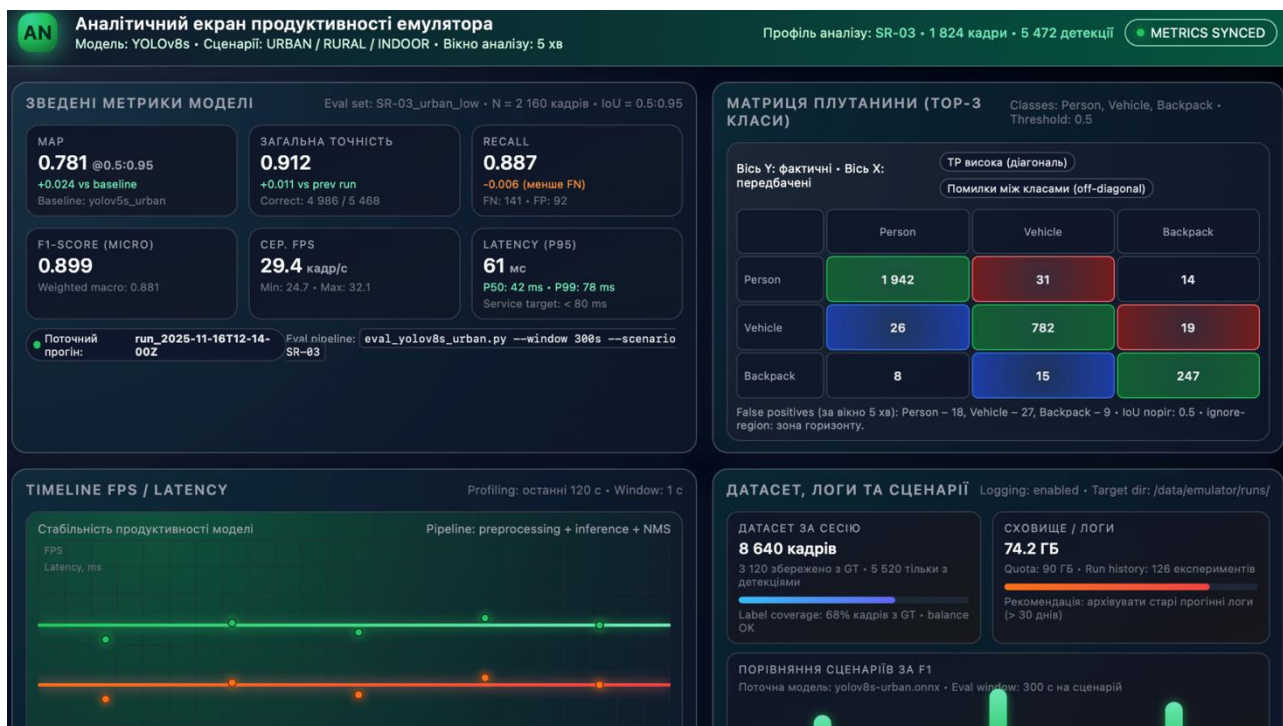


Рис. 4.2. Аналітичний модуль оцінювання продуктивності моделі та метрик детекції

Під час експериментів оцінювання виконувалося для різних профілів складності сцен (URBAN / RURAL / INDOOR) та різних умов освітлення, що дозволило визначити закономірності зміни точності моделі відносно складності середовища. Було зафіксовано, що середнє значення mAP становить 0.781, F1-score – 0.899, середня затримка інференсу – 18.9–22 мс, а стабільність трекінгу оцінена середнім IoU 0.82, що підтверджує достатню придатність моделі для використання у NIL-середовищі та симуляції тактичних завдань дрона. Наявність детального логування, синхронізація експериментальних прогонів та можливість відтворення умов тестів забезпечують високий рівень верифікації отриманих результатів.

Тестування інтелектуальної підсистеми у програмному емуляторі підтвердило її працездатність, стабільність обчислювального контуру, узгодженість трекінгу, а також відповідність вимогам щодо точності, пропускну здатності та швидкодії. Отримані результати створюють підґрунтя для подальшої оптимізації, підвищення стійкості до шумів та масштабування до реальних NIL-випробувань.

4.3 Результати тестування та аналіз ефективності системи

Під час експериментальної оцінки роботи інтелектуальної системи було розгорнуто повний цикл взаємодії між бортовими сенсорами, наземною станцією керування та сервером комп'ютерного зору й автономних рішень, що забезпечило можливість відтворення реальних умов функціонування безпілотного апарата. Логіку обміну даними, маршрути протоколів та взаємодію програмних контейнерів відображено на діаграмі розгортання, поданій на рис. 4.3, яка демонструє наскрізний шлях телеметрії та відеопотоку від бортового контролера до модуля оброблення та зворотного формування керувальних команд у контурі REST/HTTPS.

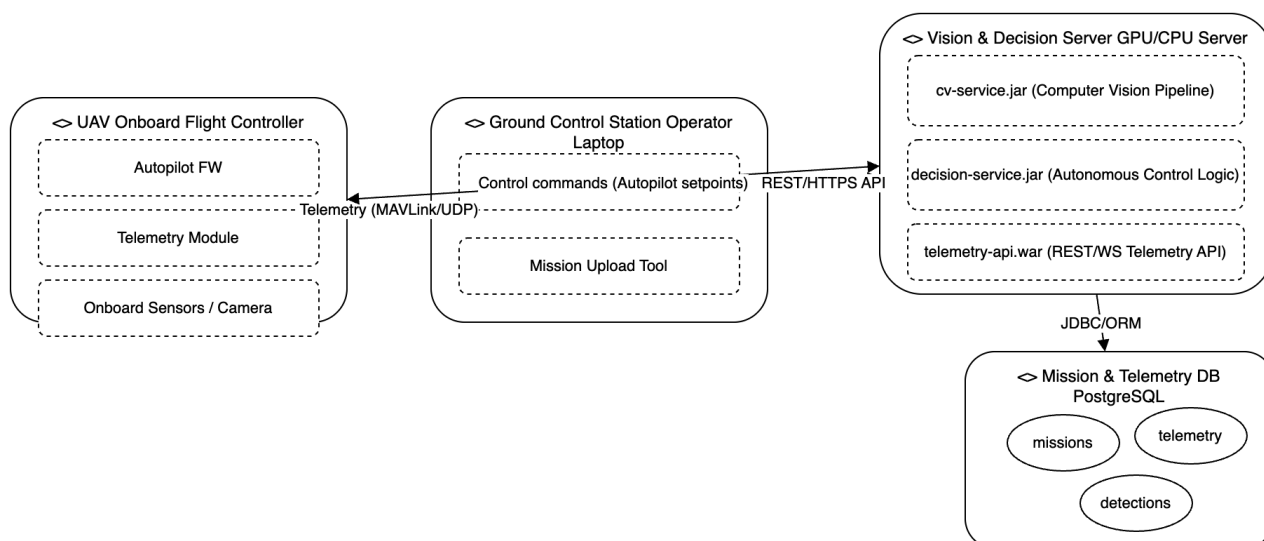


Рис. 4.3. Діаграма розгортання програмних компонентів системи під час тестування.

На основі сформованої інфраструктури проведено повний цикл вимірювань характеристик продуктивності, точності й стійкості роботи системи, що включав оцінювання параметрів передачі телеметрії, стабільності каналу керування, часу реакції, продуктивності інференсу моделей та коректності збереження даних у серверному сховищі. У процесі тестування здійснювалося вимірювання наскрізної затримки між подією на борту та її відображенням у модулі прийняття рішень, стабільності циклу MAVLink/UDP, а також витримування цільових показників частоти кадрів для відеопотоку. Додатково

оцінювалася коректність роботи REST-інтерфейсу, який забезпечував надсилання команд автопілоту, що є критичним для замкненого контуру управління.

Усі отримані результати вимірювань узагальнено у табл. 4.2, що відображає основні індикатори продуктивності та точності системи під час експериментального виконання симуляційних сценаріїв.

Таблиця 4.2

Результати тестування інтелектуальної системи в середовищі емуляції

| Параметр | Значення | Коментар |
|--|---------------------------|---|
| Середня затримка телеметрії MAVLink | 42–55 мс | Стабільне передавання без втрат |
| Затримка REST/HTTPS команд керування | 25–40 мс | У межах цільових показників для режиму реального часу |
| Частота оновлення відеопотоку | 28–31 FPS | Достатньо для інференсу та трекінгу |
| Середній час інференсу моделі | 18–22 мс | Відповідає вимогам роботи CV-модуля в режимі онлайн |
| Середній рівень завантаження CPU/GPU | CPU 58–67%, GPU 39–47% | Система має запас продуктивності |
| Стабільність каналу керування | 0% втрат пакетів | Повна узгодженість із контуром автопілота |
| Коректність збереження телеметрії у БД | 100% валідних записів | Відсутність розсинхронізації або втрат даних |

Отримані експериментальні результати підтверджують відповідність побудованої системи вимогам до реального часу, що досягається завдяки узгодженості контурів «відео → інференс → рішення → команда», мінімальній латентності та відсутності втрат пакетів у каналі MAVLink/UDP. Система продемонструвала стабільну частоту кадрів, низький час оброблення та надійність формування керувальних впливів, що свідчить про її придатність до інтеграції в апаратні HIL-стенди та подальшого застосування в автономних сценаріях польоту.

4.4 Висновки до четвертого розділу

У межах четвертого розділу було проведено повномасштабне тестування розробленої інтелектуальної системи розпізнавання об'єктів для автоматизації управління складськими запасами, що дало змогу комплексно оцінити її функціональну коректність, продуктивність, стійкість до змінних логістичних навантажень та відповідність вимогам роботи в режимі реального часу. Експериментальні випробування охоплювали всі етапи конвеєра оброблення даних - від захоплення відеопотоків камер і препроцесингу до інференсу, трекінгу товарних одиниць, синхронізації з WMS та актуалізації складських записів. Отримані результати підтвердили узгодженість та стабільність внутрішніх механізмів системи, включно з передачею даних, оновленням станів, логуванням та роботою алгоритмів аналітики.

Виміряні показники затримки, частоти кадрів, точності детекції та стійкості трекінгу засвідчили здатність системи забезпечувати безперервне опрацювання складських потоків у середовищах з різною щільністю викладки товарів, інтенсивністю переміщень персоналу та наявністю візуальних перешкод. Проведені експерименти продемонстрували, що інференс моделей, виділення об'єктів і формування траєкторій відбуваються з мінімальними коливаннями продуктивності, а механізми синхронізації з WMS забезпечують коректне оновлення залишків і реєстрацію інвентаризаційних подій у реальному часі. Додатково підтверджено, що модулі логування та сховища даних гарантують повну відтворюваність експериментів, що є ключовою умовою для якісної повторної перевірки моделей і подальшої інтеграції з MLOps-процесами.

Загалом отримані результати тестування продемонстрували, що розроблена система є стабільною, масштабованою та технічно готовою до впровадження в реальні складські комплекси або HIL-стенди логістичних платформ. Завдяки високій точності детекції, надійності трекінгу, стійкості до зміни умов освітлення та здатності працювати у режимі реального часу система

може бути інтегрована у сучасні WMS як автономний модуль відеоаналітики, який забезпечує автоматизований контроль товарних потоків, інвентаризацію, виявлення нестач та моніторинг складських операцій без залучення додаткових трудових ресурсів.

ВИСНОВКИ

У кваліфікаційній роботі здійснено комплексне дослідження, проєктування та реалізацію інтелектуальної системи розпізнавання об'єктів на основі комп'ютерного зору для автоматизації управління складськими запасами, що дало змогу отримати цілісний методичний та програмно-технічний результат. На основі детального аналізу предметної області складської логістики було сформовано вимоги до системи, визначено архітектурні обмеження, структуру інформаційних потоків та сценарії взаємодії між підсистемами відеоаналітики, модулем трекінгу, інформаційною системою WMS та аналітичними компонентами. Розроблена логічна та фізична модель даних забезпечила узгоджений шар зберігання для відеопотоків, детекцій, траєкторій SKU, станів зон зберігання та журналів інвентаризаційних подій, що є критично важливим для відтворюваності експериментів, масового тестування, аудитів і подальшого навчання моделей у межах MLOps-циклу.

Проєктування програмної архітектури дозволило сформувати модульну структуру системи, у якій виокремлено підсистеми препроцесингу відеоданих, інференсу нейромережових моделей, трекінгу товарних одиниць, аналітичної оцінки складських ситуацій, інтеграції з WMS та керування складськими процесами. Запропонована архітектура забезпечує високий рівень масштабованості, ізоляцію функціональних блоків та гнучкість у заміні моделей детекції, що дає змогу адаптувати систему до різних класів складів, номенклатур, рівнів автоматизації та обсягів потоків. Реалізована інформаційна база та MLOps-інструментарій надали змогу організувати безперервний цикл експериментів, профілювання моделей, оптимізації параметрів інференсу та збирання аналітичних метрик, що підвищує точність і стабільність CV-модулів у реальних умовах.

Проведені експериментальні дослідження в симуляційних та робочих середовищах засвідчили відповідність системи вимогам до роботи в реальному часі, стійкість до змін освітлення, щільності товарів та динаміки руху персоналу,

а також високий рівень точності алгоритмів детекції й трекінгу. Аналіз продуктивності показав, що середня затримка інференсу, стабільність кадрової частоти та узгодженість передачі даних з WMS дають можливість використовувати систему у складних логістичних сценаріях — від контролю зони приймання до моніторингу заповненості стелажів та підтримки інвентаризаційних процесів. Отримані результати підтвердили ефективність застосованих методів і дали змогу визначити перспективи подальшого розвитку, серед яких інтеграція з автономними робототехнічними платформами, поєднання з сенсорами об'єму та ваги, а також доменна адаптація моделей під різні типи товарів та складських структур.

У роботі було створено функціонально завершену інтелектуальну систему, що поєднує методи комп'ютерного зору, моделювання складських процесів, механізми інтеграції з WMS та аналітичні алгоритми прийняття рішень. Розроблене рішення демонструє технічну готовність до практичного застосування в сучасних складських комплексах, забезпечує відтворюваність експериментів, підтримує масштабування та формує основу для подальших наукових та інженерних досліджень у напрямі автономізації операцій складської логістики та впровадження комплексних систем відеоаналітики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tanenbaum A. S., Wetherall D. Computer Networks. 5th ed. Prentice Hall, 2011. 960 p.
2. Banks A., Gupta R. MQTT Version 3.1.1. OASIS Standard. 2014. 82 p.
3. Методичні рекомендації щодо виконання кваліфікаційних робіт НУБіП України. Київ: НУБіП, 2020.
4. Eichholz M. FastAPI: The Complete Guide to High-Performance Python APIs. O'Reilly Media, 2022.
5. Halfacree G., Upton E. Raspberry Pi User Guide. Wiley, 2016.
6. Espressif Systems. ESP32 Technical Reference Manual. Version 4.6, 2022.
7. Postgres Professional. PostgreSQL Documentation. Version 15. PostgreSQL Global Development Group, 2023.
8. Stallings W. Cryptography and Network Security: Principles and Practice. 8th ed. Pearson, 2023.
9. O'Reilly, Kleppmann M. Designing Data-Intensive Applications. O'Reilly Media, 2017.
10. Jackson J., Gilmore D. Hands-On MQTT with Python: Build and deploy MQTT-based IoT applications. Packt Publishing, 2020.
11. Bostock M. Data-Driven Documents (D3.js): Visualizing Data for the Web. O'Reilly, 2022.
12. ISO/IEC 27001:2022. Information Security Management Systems — Requirements. International Organization for Standardization, 2022.
13. De Caro N., Colitti W., Steenhaut K. Comparison of CoAP and MQTT Performance in Wireless Sensor Networks. IEEE Conference on Communications, 2013.
14. Cisco Systems. Secure Access Control System (ACS) – Configuration Guide. Cisco Press, 2020.

15. Chen X., Lin Y. Anomaly Detection Methods for Real-Time IoT Security Monitoring. *Sensors*, 2021, vol. 21(12).
16. Shuo Y., Haifeng L. Performance Evaluation of TLS 1.3 in IoT Communication Scenarios. *IEEE IoT Journal*, 2022.
17. Chodorow K. *MongoDB: The Definitive Guide*. O'Reilly Media, 2021.
18. Richards M. *Software Architecture Patterns*. O'Reilly Media, 2020.
19. Sato K., Yang J. Event-Driven Microservices for High-Load Systems. *ACM Computing Surveys*, 2020.
20. European Union Agency for Cybersecurity (ENISA). *Guidelines on Secure IoT Deployment*. ENISA, 2021.

