

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

«На правах рукопису»

УДК 621.375.9

До захисту допущено

Зав.кафедри _____
Андрій МОВЧАНЮК_

«__» _____ 20__ р.

Магістерська дисертація

Магдича Віталія Сергійовича

на здобуття ступеня магістра

**за освітньо-науковою програмою «Радіоелектронна інженерія»
за спеціальністю 172 «Телекомунікації та радіотехніка»
на тему: «Автоматизована система виявлення малорозмірних об'єктів»**

Виконав

студент 2 курсу, групи РЕ-41мн

Магдич Віталій Сергійович



Керівник:

Професор, д.т.н.

Степанов Михайло Миколайович



Рецензент:

Доцент кафедри прикладних інформаційних систем

Факультету інформаційних технологій

Київського національного університету ім. Тараса Шевченка

д.т.н., доцент

Прізвище, ім'я, по батькові

Фендьо Олена Миколаївна

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка) _____



Київ — 2026

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Електронні комунікації та радіотехніка »

Освітньо-наукова програма «Радіоелектронна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Андрій МОВЧАНЮК

«__» _____ 20__ р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Магдичу Віталію Сергійовичу

(прізвище ім'я по батькові)

1. Тема дисертації «**Автоматизована система виявлення малорозмірних об'єктів**»

науковий керівник дисертації — Степанов Михайло Миколайович, Професор,
Д.Т.Н.

затверджені наказом по університету від «02» квітня 2026 р. №1377-с

2. Термін подання студентом дисертації 18 травня 2026 року

3. Об'єкт дослідження — процеси автоматичного виявлення, нейромережевого розпізнавання та пасивної геолокації наземних малорозмірних об'єктів у моніторингових комплексах аеророзвідки на базі безпілотних літальних апаратів.

4. Предмет дослідження — моделі інстанс-сегментації ШІ, просторово-геометричні моделі прямого візування, алгоритми просторово-часової синхронізації та дедуплікації даних, а також інтерактивні геоінформаційні інтерфейси наземних станцій керування БПЛА.

5. Перелік завдань, які потрібно розробити

- 1) Провести системний аналіз сучасного стану, архітектурних рішень та математичних методів побудови автоматизованих комплексів повітряного моніторингу та систем комп'ютерного зору.
- 2) Розробити просторово-геометричну модель пасивної геолокації наземних об'єктів на основі проективної геометрії методом прямого візування.
- 3) Створити завадостійкий алгоритм жорсткої просторово-часової синхронізації матеріалів аерозйомки та телеметрії за бортовими часовими мітками з використанням лінійної інтерполяції.
- 4) Обґрунтувати вибір згорткової нейронної мережі.
- 5) Реалізувати тактичний дашборд (інтерфейс) оператора.
- 6) Провести експериментальні дослідження та верифікацію розробленого програмно-апаратного комплексу методами імітаційного моделювання.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу — просторово-геометрична модель визначення координат наземного об'єкта методом прямого візування; криві метрик навчання нейромережі та матриці невідповідностей класифікації; екранні форми графічного інтерфейсу (дашборду) тактичної карти оператора; результати перевірок ідентифікації техніки на основі тестовитих датасетів.

7. Орієнтовний перелік публікацій — наукові статті та тези доповідей за тематикою магістерської дисертації у фахових виданнях та матеріалах міжнародних і всеукраїнських науково-практичних конференцій молодих учених, аспірантів та студентів.

8. Дата видачі завдання 01 жовтня 2025 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Отримання теми магістерської дисертації	01.10. – 20.10. 2025р.	
2	Розробка плану магістерської дисертації	21.10. – 31.10.2025 р.	
3	Початок збору інформації для дослідження	01.11. – 15.12.2025 р.	
4	Розробка математичних моделей геолокації та алгоритмів синхронізації	16.12.2025 – 31.01.2026 р.	
5	Створення датасету та навчання моделі нейронної мережі	01.02. – 28.02.2026 р.	
6	Програмна реалізація ГІС - інтерфейсу оператора	01.03. – 31.03.2026 р.	
7	Проведення імітаційного моделювання та оцінка ефективності системи	01.04. – 30.04.2026 р.	
8	Оформлення магістерської дисертації	16.05.2026 р.	

Студент — Магдич Віталій Сергійович

Науковий керівник — Степанов Михайло Миколайович

Дисертацією є рукопис.

Робота виконана на кафедрі «Прикладної радіоелектроніки» Національного технічного університету України "Київського політехнічного інституту імені Ігоря Сікорського".

Науковий керівник: доктор технічних наук, професор Степанов Михайло Миколайович Національного технічного університету України "Київського політехнічного інституту імені Ігоря Сікорського".

Апробація результатів дисертації. Основні положення та результати дослідження доповідалися на Інтернет-конференції, матеріали якої опубліковано на платформі konferenciaonline.org.

Публікації. За матеріалами дисертації опубліковано 1 наукову працю (тези доповіді «ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АРХІТЕКТУРИ YOLO ДЛЯ ДЕТЕКЦІЇ ОБ'ЄКТІВ З БПЛА»).

ЗМІСТ

РЕФЕРАТ	7
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ	11
ВСТУП	12
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ АЕРОРОЗВІДКИ ТА ПОСТАНОВКА ЗАДАЧІ.....	15
1.1 Огляд існуючих систем аеророзвідки та їх недоліки	15
1.2 Аналіз існуючих програмних комплексів та аналогів автоматизації розвідки	17
1.3 Порівняльний аналіз архітектур нейронних мереж для детекції об'єктів.....	19
1.3 Обґрунтування вибору гібридної схеми обробки даних.....	23
1.4 Постановка задачі дослідження.....	25
Висновок до розділу	27
РОЗДІЛ 2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА АЛГОРИТМІВ ГЕОЛОКАЦІЇ	29
2.1 Математична модель визначення координат на основі проективної геометрії	29
2.2 Врахування телеметричних даних та оптичних параметрів камери .	31
2.3 Алгоритм синхронізації відеопотоку з даними телеметрії за часовими мітками	33
Висновки до розділу	35
РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ	37
3.1 Базова апаратна оснастка БПЛА (Бортовий сегмент).....	37
3.2 Архітектура наземної системи керування та збору даних (Наземний сегмент).....	48
3.3 Підготовка датасету та навчання нейронної мережі YOLO	50
3.3.1 Формування навчальної вибірки (набору даних)	51
3.3.2 Методика розмітки у середовищі Label Studio	52
3.3.3 Процес навчання нейронних мереж.....	53
3.3.4 Порівняльний аналіз результатів навчання.....	55
3.4 Розробка програмного забезпечення з інтеграцією Google Maps API	58
3.4.1 Структуризація та синхронізація вхідних телеметричних даних.....	58

3.4.2 Алгоритм інтерполяції та синхронізації кадрів	59
3.4.3 Імплементация нейромережевого трекінгу та апаратна оптимізація	60
3.4.4 Математична модель геопросторової прив'язки об'єктів	61
3.4.5 Алгоритм просторово-часової фільтрації та дедуплікації даних	61
3.5 Реалізація інтерфейсу оператора та механізмів верифікації цілей	63
3.5.1 Буферизація відеоданих та експорт доказів	63
Висновки до розділу	68
РОЗДІЛ 4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ МЕТОДАМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ	70
4.1 Опис середовища імітаційного моделювання та методики проведення експерименту	70
4.1.1 Концепція та архітектура імітаційного середовища	70
4.1.2 Структуризація та аналіз вхідних матриць телеметрії.....	72
4.1.3 Статистичний аналіз апіорної геометрії датасету за мітками анотування	73
4.1.4 Методика та критерії оцінювання ефективності комплексу	74
4.2 Дослідження точності ідентифікації в залежності від відстані до цілі	75
Висновки до розділу	80
ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТКИ.....	87
ДОДАТОК А. Лістинг коду main.py	87
ДОДАТОК Б. Схема алгоритму роботи автоматизованої системи виявлення та геолокації цілей.....	96

РЕФЕРАТ

Структура та обсяг реферату. Дисертаційна робота складається із змісту, переліку умовних позначень, вступу, 4-х розділів, висновків, списку використаних джерел. Містить: 17 – ілюстрацій, 5 – таблиць, 96 – сторінок, 2 – додатки.

Основний зміст роботи.

1. Проведено комплексний аналіз існуючих систем аеророзвідки та порівняння архітектур глибоких нейронних мереж (YOLO, SSD, Faster R-CNN) за критеріями швидкодії та точності. На основі отриманих даних обґрунтовано вибір гібридної схеми обробки інформації, де дрон виступає сенсором, а ресурсомісткі задачі детекції виконуються на наземній станції з використанням GPU, що дозволяє застосовувати більш потужні моделі ШІ без критичного навантаження на бортову електроніку та акумулятор БПЛА.
2. Розроблено математичну модель визначення географічних координат наземних об'єктів, яка базується на методах проєктивної геометрії та тригонометричних перетвореннях. Модель комплексно враховує телеметричні дані (координати GPS, барометричну та лазерну висоту, кути Ейлера: крен, тангаж, рилання) та внутрішні оптичні параметри камери (фокусна відстань, розмір матриці), що дозволяє розраховувати проєкцію центру об'єкта на земну поверхню з високою достовірністю.
3. Спроектовано архітектуру програмно-апаратного комплексу та реалізовано алгоритм синхронізації різнорідних потоків даних (відео та телеметрії) за часовими мітками. Це рішення дозволяє нівелювати затримки передачі сигналу та співставити конкретний кадр відеопотоку з точним просторовим положенням дрона в момент зйомки, що є критичним фактором для мінімізації похибки геолокації цілі.

4. Виконано навчання згорткової нейромережі на базі архітектури YOLO (You Only Look Once) з використанням власного анотованого датасету військової техніки. Завдяки застосуванню методів аугментації даних (геометричні спотворення, зміна освітлення, додавання шуму) та трансферного навчання, досягнуто високих показників точності (Precision) та повноти (Recall), що забезпечує надійну ідентифікацію об'єктів навіть у складних умовах спостереження.
5. Розроблено клієнтське програмне забезпечення з інтеграцією Google Maps API для автоматичного нанесення маркерів виявлених цілей на інтерактивну карту в режимі реального часу. Інтерфейс включає інструменти підтримки прийняття рішень: спливаючі вікна з відеофрагментами для верифікації об'єкта оператором, історію подій та модуль експорту координат для передачі суміжним підрозділам.
6. Методом імітаційного моделювання досліджено роботу системи у віртуальному середовищі на висотах польоту від 50 до 200 м. Оцінено середньоквадратичну похибку позиціонування та встановлено її залежність від кута візування камери. Також на синтетичних тестових вибірках підтверджено ефективність алгоритмів розпізнавання та коректність роботи логіки програмного комплексу без ризику втрати дороговартісного обладнання.

Ключові слова: АЕРОРОЗВІДКА, БПЛА, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННА МЕРЕЖА, ГЕОЛОКАЦІЯ, ВИЗНАЧЕННЯ КООРДИНАТ, API, АВТОМАТИЗОВАНА СИСТЕМА.

ABSTRACT

Structure and scope of the thesis. The thesis consists of a table of contents, a list of abbreviations, an introduction, 4 chapters, conclusions, and a list of references. It contains: 17 illustrations, 5 tables, 96 pages and 2 appendices.

Main content.

1. A comprehensive analysis of existing aerial reconnaissance systems and a comparison of deep neural network architectures (YOLO, SSD, Faster R-CNN) based on speed and accuracy criteria were conducted. Based on the obtained data, the choice of a hybrid information processing scheme was justified, where the drone acts as a sensor, and resource-intensive detection tasks are performed on a ground station using a GPU. This allows the use of more powerful AI models without critical load on the onboard electronics and the UAV battery.
2. A mathematical model for determining the geographic coordinates of ground objects was developed, based on methods of projective geometry and trigonometric transformations. The model comprehensively takes into account telemetry data (GPS coordinates, barometric and laser altitude, Euler angles: roll, pitch, yaw) and internal optical camera parameters (focal length, sensor size), which allows calculating the projection of the object's center onto the earth's surface with high reliability.
3. The architecture of the hardware-software complex was designed, and an algorithm for synchronizing heterogeneous data streams (video and telemetry) by timestamps was implemented. This solution allows mitigating signal transmission delays and matching a specific video frame with the exact spatial position of the drone at the moment of shooting, which is a critical factor for minimizing the target geolocation error.
4. Training of a convolutional neural network based on the YOLO (You Only Look Once) architecture was performed using a custom annotated dataset of

military equipment. Thanks to the application of data augmentation methods (geometric distortions, lighting changes, noise addition) and transfer learning, high indicators of precision and recall were achieved, ensuring reliable object identification even in complex observation conditions.

5. Client software with Google Maps API integration was developed for automatically placing markers of detected targets on an interactive map in real-time. The interface includes decision support tools: pop-up windows with video fragments for object verification by the operator, an event history, and a coordinate export module for data transfer to adjacent units.
6. Using the method of simulation modeling, the system's operation was investigated in a virtual environment at flight altitudes from 50 to 200 m. The root-mean-square positioning error was estimated, and its dependence on the camera viewing angle was established. Also, on synthetic test samples, the effectiveness of recognition algorithms and the correctness of the software complex logic were confirmed without the risk of losing expensive equipment.

Keywords: AERIAL RECONNAISSANCE, UAV, COMPUTER VISION, NEURAL NETWORK, GEOLOCATION, COORDINATE DETERMINATION, API, AUTOMATED SYSTEM.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

API – (Application Programming Interface) прикладний програмний інтерфейс

GPS – (Global Positioning System) глобальна система позиціонування

SSD – (Single Shot MultiBox Detector) метод одноразового детектування об'єктів

YOLO – (You Only Look Once) архітектура нейронної мережі для детекції об'єктів у реальному часі

БПЛА – безпілотний літальний апарат

ЗНМ – згорткова нейронна мережа (Convolutional Neural Network, CNN)

КЗ – комп'ютерний зір. ПЗ – програмне забезпечення

ПК – персональний комп'ютер (наземна станція керування)

ШІ – штучний інтелект

ТТХ – тактико-технічні характеристики

mAP – (mean Average Precision) усереднена середня точність (комплексна метрика оцінки нейромереж)

ВСТУП

Актуальність теми. У сучасних умовах ведення бойових дій безпілотні літальні апарати (БПЛА) стали ключовим елементом розвідки. Проте ефективність їх використання часто обмежується людським фактором: ручний пошук цілей на відеопотоці вимагає значної концентрації уваги, що призводить до швидкої втоми оператора та можливого пропуску об'єктів. Крім того, процес визначення точних координат виявленої техніки та нанесення їх на карту займає дорогоцінний час. Автоматизація цих процесів шляхом поєднання технологій комп'ютерного зору (ШІ) та алгоритмів геолокації дозволяє значно скоротити цикл «виявлення-ураження» та підвищити ситуаційну обізнаність, що робить тему дисертації актуальною.

Мета і задачі дослідження. Метою роботи є підвищення ефективності аеророзвідки шляхом розробки автоматизованої системи, яка забезпечує виявлення, ідентифікацію та визначення географічних координат наземної техніки в реальному часі з подальшою візуалізацією на інтерактивній карті.

Для досягнення мети вирішуються такі **задачі**:

1. Провести аналіз методів комп'ютерного зору та існуючих підходів до геолокації об'єктів з борта БПЛА.
2. Розробити математичну модель розрахунку координат цілі на основі даних телеметрії (GPS, висота, кути орієнтації) та параметрів камери.
3. Спроекувати архітектуру програмно-апаратного комплексу та обґрунтувати вибір компонентів.
4. Здійснити навчання нейронної мережі для розпізнавання військової техніки у відеопотоці.
5. Розробити клієнтське програмне забезпечення для візуалізації обстановки на базі Google Maps API.

6. Дослідити точність позиціонування та якість розпізнавання шляхом імітаційного моделювання.

Об'єкт дослідження. Процес автоматизованого отримання та обробки розвідувальної інформації з використанням безпілотних літальних апаратів.

Предмет дослідження. Методи та алгоритми комп'ютерного зору для ідентифікації наземних об'єктів та математичні моделі визначення їх географічних координат на основі комплексування навігаційних та оптичних даних.

Методи дослідження. У роботі використано: методи аналізу та синтезу (для огляду існуючих рішень); методи проєктивної геометрії та тригонометрії (для розробки моделі геолокації); методи машинного навчання та теорії нейронних мереж (для детекції об'єктів); об'єктно-орієнтоване програмування (для реалізації ПЗ); методи імітаційного моделювання (для експериментальної перевірки точності системи в середовищі емуляції).

Наукова новизна одержаних результатів:

1. **Удосконалено** метод визначення координат наземних цілей за монокулярним зображенням, який, на відміну від існуючих, комплексно враховує дані лазерного висотоміра та повний набір кутових параметрів орієнтації камери і дрона, що дозволяє підвищити точність геолокації без використання активних далекомірів.
2. **Набуло подальшого розвитку** застосування згорткових нейронних мереж у системах тактичної розвідки шляхом створення гібридної архітектури обробки даних, що забезпечує автоматичну прив'язку розпізнаного об'єкта до геоінформаційної системи в режимі реального часу.

Практичне значення одержаних результатів. Розроблено діючий прототип програмно-апаратного комплексу, який дозволяє оператору

отримувати верифіковані дані про розташування ворожої техніки з мінімальною затримкою. Створене програмне забезпечення з інтеграцією Google Maps API спрощує орієнтування на місцевості та прийняття рішень. Результати роботи можуть бути використані при розробці нових зразків розвідувальних БПЛА та тренажерних комплексів для навчання операторів.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ АЕРОРОЗВІДКИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд існуючих систем аеророзвідки та їх недоліки

На сучасному етапі ведення бойових дій безпілотні літальні апарати (БПЛА) стали невіддільним та ключовим елементом тактичної та оперативної розвідки. Вони забезпечують безперервний моніторинг поля бою, коригування артилерійського вогню та виявлення замаскованих позицій противника. Залежно від радіуса дії, часу автономного польоту та цільового призначення, наявний парк розвідувальних дронів можна умовно поділити на дві основні категорії: комерційні дрони (адаптовані для військових потреб) та спеціалізовані тактичні комплекси.



А)



Б)

Рисунок 1.1 — Приклади комерційних дронів та спеціалізованих тактичних комплексів (А – DJI Mavic 3 Pro, Б – Лелека - 100)

Комерційні дрони (коптерного типу). Широкого поширення на лінії зіткнення набули комерційні квадрокоптери, такі як лінійки DJI (Mavic 3, Matrice 30T/300/350 RTK) та Autel (EVO II, EVO Max). Їхня головна перевага полягає у відносній дешевизні, доступності, простоті розгортання та здатності зависати над об'єктом для детального огляду. Сучасні моделі оснащуються якісними тепловізійними матрицями та камерами з оптичним збільшенням.

Проте, оскільки ці платформи початково створювалися для цивільного сектору (інспекція інфраструктури, зйомка), вони мають суттєві недоліки у контексті військового застосування:

- Відсутність автоматизованої аналітики: Базове програмне забезпечення цих дронів не містить вбудованих інструментів для автоматичного виявлення військової техніки. Уся відповідальність за пошук цілей лежить виключно на зоровому апараті оператора.
- Складність геолокації: Більшість легких коптерів не оснащені активними лазерними далекомірами. Оператору доводиться візуально співставляти орієнтири з відеопотоку з топографічною картою, щоб приблизно визначити координати цілі, що займає час і призводить до значних похибок.

Тактичні БПЛА (літакового типу). Для глибинної розвідки застосовуються спеціалізовані безпілотні авіаційні комплекси літакового типу, такі як вітчизняні «Лелека-100», «Шарк» (SHARK UAV), «Фурія» або PD-2. Вони здатні перебувати в повітрі по декілька годин, працювати в умовах інтенсивної дії засобів РЕБ та вести розвідку на відстанях у десятки кілометрів.

Ці системи оснащені потужними гіростабілізованими оптоелектронними системами (підвісами) із зумом великого збільшення та тепловізорами з високою роздільною здатністю. Вони здатні автоматично обчислювати координати центру кадру на основі телеметричних даних дрона та кутів повороту камери.

Однак, незважаючи на технологічну перевагу, процес виявлення цілей залишається вузьким місцем:

- Напівавтоматичний режим: Сучасні підвіси мають функцію автосупроводження цілі (target tracking), але вона активується лише після того, як оператор самостійно знайде об'єкт на екрані та «захопить» його в рамку.
- Надлишок інформації: Враховуючи велику висоту польоту та специфіку місцевості, оператор наземної станції змушений годинами аналізувати

одноманітний ландшафт у пошуках замаскованої техніки, що робить цей процес неефективним без залучення систем машинного зору.

1.2 Аналіз існуючих програмних комплексів та аналогів автоматизації розвідки

Стрімкий розвиток безпілотних технологій призвів до появи на ринку значної кількості програмно-апаратних рішень, спрямованих на підвищення ефективності аеророзвідки. Для визначення доцільності та новизни розроблюваної системи необхідно провести аналіз існуючих аналогів. Усі поточні рішення можна умовно поділити на три основні категорії: національні системи ситуаційної обізнаності, західні інтегровані платформи на базі штучного інтелекту та комерційні (цивільні) системи управління БПЛА.

1. Національні системи ситуаційної обізнаності та управління боєм У сучасних реаліях ведення бойових дій в Україні широко застосовуються такі системи ситуаційної обізнаності як «Кропива», «Delta», «Gis Arta» та «Armor».

- Переваги: Ці системи є надзвичайно ефективними інструментами для артилерійських підрозділів. Вони забезпечують єдине інформаційне середовище, дозволяють обмінюватися даними між підрозділами та швидко розраховувати балістичні поправки.
- Недоліки (Обмеження): Головним недоліком цих систем у контексті аеророзвідки є відсутність вбудованих модулів комп'ютерного зору для автоматичної обробки «сирого» відеопотоку з БПЛА. Інтеграція даних відбувається здебільшого в ручному або напівавтоматичному режимі. Оператор дрона змушений візуально (за допомогою власних очей) ідентифікувати замасковану техніку на екрані, після чого вручну вводити її координати або ставити маркер у системі (наприклад, у «Кропиві»). Це не вирішує описану раніше проблему когнітивного перевантаження оператора та залишає високим ризик людської помилки.

2. Західні військові платформи з інтеграцією штучного інтелекту Передові оборонні компанії світу вже розробляють системи, що поєднують БПЛА та ШІ. Яскравими прикладами є програмне забезпечення «Lattice» від компанії Anduril, екосистема «Gotham» від Palantir або ініціативи на кшталт Project Maven (США).

- Переваги: Ці платформи забезпечують найвищий рівень автоматизації. Вони використовують складні алгоритми злиття даних (sensor fusion) та комп'ютерного зору для автоматичного виявлення, класифікації та супроводження цілей у режимі реального часу, майже повністю виключаючи людину з процесу пошуку.
- Недоліки (Обмеження): Основним бар'єром для їх використання є закритість архітектури (пропріетарність) та надзвичайно висока вартість. Такі системи зазвичай постачаються як єдиний апаратно-програмний комплекс і вимагають використання специфічних, дороговартісних військових сенсорів. Вони не пристосовані для масового розгортання на рівні тактичних підрозділів, які здебільшого використовують комерційні дрони або БПЛА волонтерського виробництва.

3. Комерційні (цивільні) системи управління екосистемами БПЛА Виробники комерційних дронів також пропонують власне програмне забезпечення для аналізу даних, наприклад DJI FlightHub 2 або Autel Enterprise.

- Переваги: Вони мають вбудовані функції картографування, синхронізації відео з кількох дронів та базові алгоритми розпізнавання об'єктів (наприклад, функція Smart Track).
- Недоліки (Обмеження): Базові нейронні мережі комерційних дронів натреновані виключно на цивільні об'єкти (цивільні автомобілі, люди, будівлі). Вони не здатні ефективно розпізнавати специфічну військову техніку, тим паче в умовах застосування засобів маскуванню

(камуфляжні сітки, природне укриття). Крім того, ці системи не дозволяють експортувати точні географічні координати виявлених цілей у сторонні військові ГІС без складних неофіційних модифікацій.

Аналіз існуючих рішень показує наявність суттєвої прогалини у технологічному забезпеченні тактичної ланки. З одного боку існують ефективні ГІС («Кропива», «Delta»), яким бракує автоматизації візуального пошуку. З іншого — існують потужні західні AI-системи, які є закритими та недоступними для масового застосування з наявним парком комерційних БПЛА.

1.3 Порівняльний аналіз архітектур нейронних мереж для детекції об'єктів

Як зазначалося раніше, основним способом підвищення ефективності тактичної аеророзвідки є автоматизація процесу пошуку цілей за допомогою технологій комп'ютерного зору. Для успішного впровадження цього методу надзвичайно важливо обрати оптимальну архітектуру штучної нейронної мережі. Вона має забезпечувати ефективну роботу в умовах обмежених обчислювальних ресурсів і здатність до обробки відеопотоку з БПЛА в режимі реального часу.

Сучасні підходи до детекції об'єктів на базі глибокого навчання (Deep Learning) здебільшого поділяються на дві основні категорії: двостадійні детектори (на основі пропозиції регіонів, region proposal approaches) та одностадійні детектори (без пропозиції регіонів, non-region proposal approach).

Двостадійні методи спочатку генерують пропозиції регіонів (області зображення, де з високою ймовірністю знаходиться об'єкт), а потім класифікують ці регіони та уточнюють їхні координати. До таких методів належать архітектури R-CNN, Fast R-CNN та Faster R-CNN.

Одностадійні методи розглядають детекцію як єдину задачу регресії, прогнозуючи обмежувальні рамки (bounding boxes) та ймовірності класів за

один прохід через нейронну мережу. До цієї категорії належать алгоритми Single Shot Detector (SSD) та сімейство You Only Look Once (YOLO)[5].

У даному підрозділі буде проведено порівняльний аналіз швидкодії та точності трьох найпопулярніших представників цих категорій: Faster R-CNN, SSD та YOLO, з метою визначення найбільш ефективного рішення для задачі виявлення цілей у різних умовах навколишнього середовища.

Для об'єктивного аналізу ефективності архітектур нейронних мереж у задачах комп'ютерного зору зазвичай застосовують набір стандартизованих кількісних метрик. Основними серед них є точність (Precision), повнота (Recall), гармонійне середнє цих показників (F1-оцінка) та усереднена середня точність (mean Average Precision, mAP). Ці метрики забезпечують всебічну оцінку здатності моделі не лише правильно класифікувати об'єкти, але й з високою надійністю визначати їхнє просторове розташування через обмежувальні рамки.

Згідно з результатами комплексного порівняльного дослідження методів глибокого навчання для детекції об'єктів у реальному часі [1], архітектури Faster R-CNN, YOLO та SSD демонструють істотні відмінності в ефективності. У дослідженні було протестовано ці моделі на репрезентативному та складному наборі даних COCO (Common Objects in Context), використовуючи різні порогові значення перетину (IoU — Intersection over Union). Зокрема, для аналізу просторової точності алгоритмів застосовували метрики $mAP@0.5$ і $mAP@0.75$. Метрика $mAP@0.5$ відображає середню точність, якщо площа перетину передбаченої обмежувальної рамки (bounding box) із еталонною складає щонайменше 50%. Це базовий стандарт для підтвердження успішного виявлення об'єкта. Водночас $mAP@0.75$ є значно суворішим критерієм, що вимагає щонайменше 75% збігу площ рамок. Такий підхід дозволяє об'єктивно оцінити здатність алгоритмів до більш точної геометричної локалізації об'єктів, що має критичне значення, наприклад, у визначенні координат під час аеророзвідки. Сукупні результати оцінки здатності

алгоритмів до точного виявлення, локалізації та класифікації представлені у Таблиці 1.1.

Таблиця 1.1 — Порівняння метрик точності детекторів об'єктів

Архітектура	Точність (Precision)	Повнота (Recall)	F1 - Оцінка	mAP@0.5	mAP@0.75
Faster R-CNN	0.85	0.78	0.81	0.75	0.62
YOLO	0.82	0.75	0.78	0.72	0.58
SD	0.78	0.72	0.74	0.68	0.54

Згідно з даними, наведеними у Таблиці 1.1, архітектура Faster R-CNN демонструє найвищі показники точності локалізації та класифікації, досягаючи значення Precision 0.85 та F1-оцінки 0.81. Алгоритм YOLO показує конкурентоспроможні результати (Precision 0.82), утримуючи баланс між точністю та швидкістю. Водночас алгоритм SSD демонструє найнижчі показники точності (0.78) та середньої точності (mAP@0.5 на рівні 0.68), що є критичним недоліком для задач аеророзвідки.

Незважаючи на найвищі показники класифікаційної точності (mAP), які демонструє двостадійна архітектура Faster R-CNN на статичних зображеннях, абстрактна точність не є вичерпним критерієм для систем тактичної аеророзвідки. Специфіка роботи БПЛА висуває додаткові жорсткі вимоги до обробки даних: необхідність безперервного аналізу динамічного відеопотоку в умовах швидкої зміни ракурсу та фону.

У цьому контексті ключовим параметром стає швидкість роботи алгоритму, що визначається кількістю оброблених кадрів за секунду (FPS — Frames Per Second). Якщо нейронна мережа не може аналізувати зображення в режимі реального часу, відповідно до швидкості надходження кадрів від камери дрона, система матиме затримку в обробці даних. Для рухомого безпілота це означає, що на момент завершення аналізу його просторове

положення може змінитися, через що отримані географічні координати цілі можуть виявитися неточними. Ще одним важливим аспектом надійності є несприйнятливність алгоритму до хибних спрацьовувань (False Positives, FP), коли елементи ландшафту помилково ідентифікуються як військова техніка. Такі помилки створюють додаткове навантаження на оператора та негативно впливають на загальну ефективність системи.

Для об'єктивного порівняння цих критичних параметрів було проаналізовано результати прикладного дослідження [2], у якому оцінювалася практична ефективність архітектур Faster R-CNN, YOLO та SSD виключно у задачах детекції об'єктів. Алгоритми тестувалися на відеокадрах високої роздільної здатності, отриманих з дронів у різних умовах навколишнього середовища. Зведені результати щодо швидкодії, кількості правильно (TP) та хибно (FP) виявлених цілей наведено у таблиці 1.2.

Таблиця 1.2 — Порівняння швидкодії та надійності архітектур нейронних мереж у задачах детекції

Метод (Роздільна здатність)	Швидкодія (FPS)	Правильно виявлені цілі (TP)	Хибні спрацьовування (FP)	F1-оцінка
Faster R-CNN (1000)	7.69	1779	0	0.54
Faster R-CNN (300)	27.18	88	0	0.03
SSD (300)	27.93	662	677	0.217
YOLO (832)	45.10	562	0	0.211
YOLO (544)	87.22	203	0	0.08

Результати практичних експериментів виявляють критичні обмеження розглянутих архітектур. Двостадійна мережа Faster R-CNN при обробці зображень високої роздільної здатності забезпечує високу кількість правильно виявлених цілей (TP = 1779), проте її швидкодія становить лише 7.69 кадрів на

секунду (FPS). Це унеможлиблює її використання для обробки відеопотоку у режимі реального часу. Збільшення швидкості за рахунок зменшення вхідного розміру (Faster R-CNN 300) призводить до катастрофічного падіння F1-оцінки до 0.03.

Алгоритм SSD (300), хоча і працює зі швидкістю 27.93 FPS, генерує неприпустимо високу кількість хибних спрацьовувань (FP = 677). Для військових застосувань високий рівень хибних ідентифікацій робить систему неефективною.

Натомість сімейство алгоритмів YOLO (зокрема YOLO 832) досягає продуктивності в 45.10 FPS, що повністю задовольняє вимоги режиму реального часу, при цьому демонструючи високу розпізнавальну здатність із нульовим рівнем хибних спрацьовувань (FP = 0).

Аналіз наведених порівняльних даних свідчить, що архітектура YOLO добре поєднує високу обчислювальну швидкість із точністю детекції. Завдяки високому FPS та низькій чутливості до хибних спрацьовувань ця модель ідеально підходить для інтеграції в автоматизовані системи виявлення наземних цілей безпілотних літальних апаратів. Враховуючи ці характеристики, для подальшої програмно-апаратної реалізації комплексу вибрано саме архітектуру YOLO.

1.3 Обґрунтування вибору гібридної схеми обробки даних

Успішне застосування нейронної мережі YOLO для виявлення наземних цілей у режимі реального часу, вимагає значних обчислювальних потужностей, зокрема використання сучасних графічних процесорів (GPU). У розробці автоматизованих систем для безпілотних літальних апаратів існують два концептуальні підходи до обробки інформації: бортова обробка (Edge Computing) та обробка на наземній станції керування (Ground Station Computing).

Обмеження бортової обробки даних. Інтеграція обчислювального модуля (наприклад, мікрокомп'ютерів серії NVIDIA Jetson або Raspberry Pi) безпосередньо на борт БПЛА має низку критичних недоліків для тактичної аеророзвідки:

1. Енергоспоживання: Робота графічних співпроцесорів під час інференсу (виведення) нейронної мережі споживає значну кількість електроенергії. Це створює додаткове навантаження на бортовий акумулятор та суттєво зменшує загальний час автономної роботи дрона.
2. Масо-габаритні характеристики: Додаткові обчислювальні модулі та необхідні для них системи активного охолодження збільшують корисне навантаження на безпілотник, що неодмінно погіршує його льотні характеристики.
3. Обмеження точності моделей: Через апаратні обмеження бортових мікрокомп'ютерів розробники змушені використовувати максимально спрощені версії нейронних мереж (наприклад, моделі з індексом "Nano" або "Tiny"). Такі моделі забезпечують необхідну швидкодію, але мають значно нижчу точність розпізнавання дрібних об'єктів порівняно з повноцінними алгоритмами, що є неприпустимим при розвідці з висоти.

З огляду на зазначені обмеження, у даній роботі запропоновано та обґрунтовано використання гібридної архітектури обробки даних.

Суть цього підходу полягає у функціональному розділенні завдань між бортовим та наземним сегментами комплексу:

- Бортовий сегмент (БПЛА) виконує виключно роль розширеної сенсорної платформи. Його завдання зводяться до збору відеоінформації, фіксації поточних телеметричних даних (географічних координат, висоти, кутів просторової орієнтації камери та дрона) і передачі цього єдиного потоку по захищеному радіоканалу на землю з мінімальною затримкою.

- Наземний сегмент (Станція керування) бере на себе всі ресурсомісткі задачі. Отриманий відеопотік синхронізується з телеметрією та обробляється потужною відеокартою (GPU) наземного комп'ютера оператора.

Такий підхід дозволяє використовувати потужні й високоточні моделі штучного інтелекту без будь-якого впливу на енергоефективність дрона. Окрім цього, наземна станція значно легше піддається модернізації у разі впровадження нових, більш вимогливих алгоритмів комп'ютерного зору. Обчислення географічних координат цілей (геоприв'язка) також здійснюється на наземній станції, що дає змогу оперативно відображати виявлені об'єкти на інтерактивній карті (ГІС).

Отже, гібридна архітектура є найбільш раціональним рішенням, яке гарантує оптимальний баланс між льотними характеристиками БПЛА та високою точністю й швидкістю системи розпізнавання.

1.4 Постановка задачі дослідження

Ми розглянули, як зараз працює тактична аеророзвідка, які програми вже існують і як виглядає будова глибоких нейронних мереж. Висновок: треба створити одну автоматичну систему, яка буде виявляти наземні цілі і визначати координати наземних цілей. Основна проблема – людська втома під час візуального пошуку. Друга проблема – закриті і дорогі західні платформи штучного інтелекту. Третя проблема – довгий час між ідентифікацією об'єкта і розрахунком координат об'єкта. Потрібно усунути ці проблеми, щоб система працювала швидко і без помилок.

Для вирішення цих проблем та досягнення загальної мети магістерської дисертації необхідно виконати наступні науково-практичні задачі:

1. Розробити математичну модель геолокації: Створити алгоритм розрахунку точних географічних координат наземної цілі (військової техніки) за монокулярним зображенням з камери БПЛА. Модель

повинна комплексно враховувати дані телеметрії дрона (координати GPS, барометричну та лазерну висоту, кути просторової орієнтації) та оптичні параметри камери.

2. Спроекувати програмно-апаратну архітектуру системи: Деталізувати структуру розробленого гібридного підходу, обґрунтувати вибір базової апаратної оснастки БПЛА (бортовий сегмент) та розробити алгоритм синхронізації різнорідних потоків даних (відео та телеметрії) за часовими мітками для їх спільної обробки на наземній станції.
3. Реалізувати модуль розпізнавання цілей: Підготувати спеціалізований набір даних (датасет) із зображеннями військової техніки та виконати навчання згорткової нейронної мережі на базі обраної архітектури YOLO для забезпечення розпізнавання об'єктів у відеопотоці в режимі реального часу без критичних затримок.
4. Розробити клієнтське програмне забезпечення (ГІС): Створити інтерактивний інтерфейс оператора наземної станції з інтеграцією картографічного сервісу (наприклад, Google Maps API) для автоматизованого нанесення виявлених цілей на карту та реалізувати механізми їх візуальної верифікації.
5. Провести експериментальне дослідження системи: Здійснити перевірку розроблених алгоритмів методами імітаційного моделювання у віртуальному середовищі. Оцінити середньоквадратичну похибку позиціонування залежно від висоти польоту та кута візування, а також підтвердити ефективність розпізнавання на синтетичних тестових вибірках.

У цій магістерській дисертації не буде розроблятися сам безпілотний літальний апарат. Не планується фізичне конструювання, збірка чи аеродинамічне проектування дрону. Робота зосереджена лише на створенні інформаційно-обчислювальної системи. Робота включає інтеграцію

алгоритмів комп'ютерного зору, математичних моделей геолокації і програмного забезпечення наземної станції. Для практичної реалізації бортового сегмента ми проаналізуємо готові моделі БПЛА, які вже є на ринку. Ми підберемо і пояснимо вибір цих моделей і відповідного бортового сенсорного обладнання. Бортове сенсорне обладнання включає оптико-електронні модулі, лазерні далекоміри і польотні контролери. Технічні характеристики цих модулів повністю задовольняють вимоги гібридній архітектури обробки даних. Я вірю, що вирішення цих задач дозволить створити працюючий прототип простої і корисної системи. Система автоматизує збір розвідувальної інформації на тактичному рівні.

Висновок до розділу

У першому розділі проведено комплексний аналіз сучасних методів та засобів тактичної аеророзвідки за допомогою БПЛА. Встановлено, що поточні підходи значною мірою покладаються на ручний візуальний пошук цілей, що в умовах високого когнітивного навантаження неминуче призводить до швидкої втоми оператора та високого ризику пропуску замаскованих об'єктів. Аналіз існуючих програмних аналогів виявив критичну прогалину між поширеними національними системами ситуаційної обізнаності, які потребують ручного введення даних, та наддорогими закритими західними платформами на базі штучного інтелекту. Це підтверджує високу актуальність розробки власної автоматизованої та доступної системи.

Для реалізації модуля комп'ютерного зору було здійснено порівняльний аналіз популярних архітектур глибоких нейронних мереж (Faster R-CNN, SSD, YOLO). На основі аналізу результатів сучасних досліджень доведено, що для задачі обробки відеопотоку з БПЛА в режимі реального часу найдоцільнішим є використання архітектури сімейства YOLO. Вона забезпечує необхідну швидкодію (понад 45 кадрів на секунду) без критичних затримок та демонструє високу стійкість до хибних спрацьовувань при розпізнаванні малорозмірних цілей з висоти.

Для подолання апаратних обмежень безпілотних платформ, таких як обмежені енергоресурси та вантажопідйомність, запропоновано і обґрунтовано застосування гібридної архітектури обробки даних. У межах цього підходу безпілотний літальний апарат (БПЛА) функціонує виключно як сенсорна платформа, тоді як усі ресурсоємні задачі, пов'язані з інференсом нейронної мережі та обчисленням просторових координат, передаються на обчислювальні потужності наземної станції управління, оснащеної GPU.

На основі виявлених проблем і запропонованих технологічних рішень сформовано загальну мету та завдання дослідження. Встановлено, що подальша робота має бути зосереджена на розробці математичних моделей пасивної геолокації, тренуванні нейронної мережі, створенні програмного забезпечення для наземної станції з інтеграцією ГІС, а також на експериментальній перевірці комплексу в умовах імітаційного моделювання.

РОЗДІЛ 2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА АЛГОРИТМІВ ГЕОЛОКАЦІЇ

2.1 Математична модель визначення координат на основі проективної геометрії

Для автоматизації процесу геолокації малорозмірних об'єктів у розроблюваній системі використовується метод прямого візування. Математична модель базується на припущенні, що після виявлення об'єкта нейронною мережею оператор або автоматизована система управління підвісом виконує центрування камери таким чином, щоб ціль була точно розташована на головній оптичній осі камери.

Такий підхід дозволяє суттєво спростити розрахунковий алгоритм, оскільки положення цілі в центрі кадру нівелює вплив оптичної дисторсії лінз та кута крену (Roll) самого безпілотного літального апарата (БПЛА) на результат обчислень. За цих умов задача визначення географічних координат зводиться до розв'язання прямокутного трикутника у просторі, де вершинами є точка знаходження БПЛА, проєкція БПЛА на земну поверхню та сама ціль.

Для побудови моделі використовуються наступні параметри телеметрії, що фіксуються бортовими сенсорами в момент наведення на об'єкт:

- X_1, Y_1 — поточні географічні координати БПЛА (отримані з системи GPS/GNSS);
- h — істинна висота польоту над рівнем цілі (виміряна лазерним далекоміром або барометричним висотоміром);
- α — кут між вертикальною віссю БПЛА та оптичною віссю камери (лінією візування);
- β — істинний азимут напрямку камери (кут рискання відносно географічної півночі).

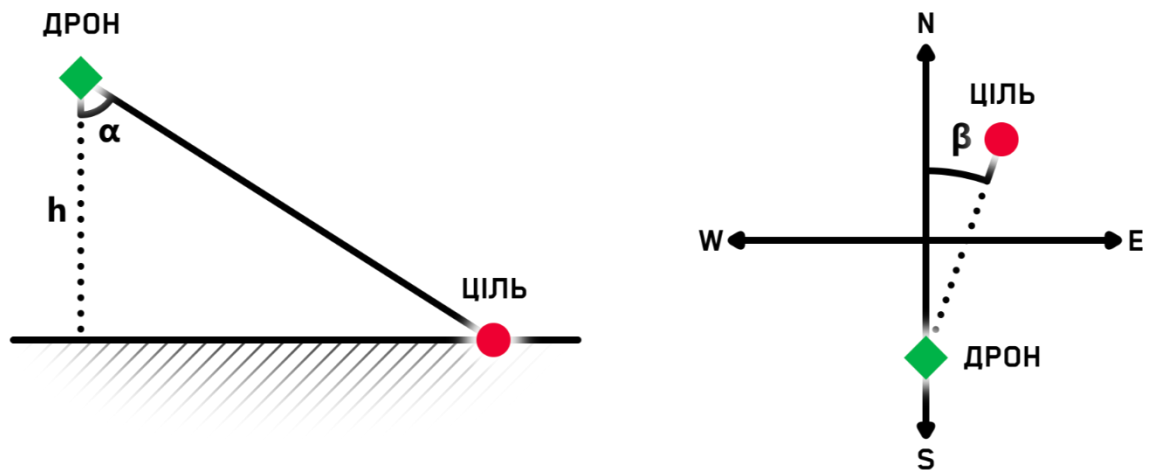


Рисунок 2.1 — Просторово-геометрична модель пасивної геолокації об'єкта методом прямого візування

Процес розрахунку координат цілі (X_2 , Y_2) складається з двох етапів. На першому етапі обчислюється горизонтальна дальність L , що є проєкцією лінії візування камери на земну площину:

$$L = h \times \tan(\alpha) \quad (2.1)$$

На другому етапі, використовуючи отримане значення дальності L та кут азимута β , розраховується просторове зміщення цілі відносно БПЛА за осями координат та визначаються її фінальні координати:

$$X_2 = X_1 + L \times \sin(\beta) \quad (2.2)$$

$$Y_2 = Y_1 + L \times \cos(\beta) \quad (2.3)$$

Дана модель забезпечує високу точність позиціонування за умови стабільного утримання цілі в центрі кадру. Оскільки алгоритм оперує безпосередньо фізичними величинами телеметрії, він демонструє високу швидкодію, що є критичним фактором для роботи системи в режимі реального часу. Використання такої моделі дозволяє мінімізувати час обробки інформації та передавати верифіковані координати в геоінформаційні системи (ГІС) з мінімальною затримкою.

2.2 Врахування телеметричних даних та оптичних параметрів камери

Для практичної реалізації математичної моделі прямого візування необхідно забезпечити безперервну обробку телеметричного потоку, що надходить від польотного контролера БПЛА та модуля керування гіростабілізованим підвісом камери. Ключовим завданням цього етапу є коректне визначення істинної висоти польоту h та абсолютних кутів просторової орієнтації лінії візування α (тангаж) та β (азимут).

Визначення істинної висоти польоту (h)

Стандартні навігаційні системи (GPS-приймачі та барометричні датчики), якими оснащуються БПЛА, зазвичай надають абсолютну висоту над рівнем моря (AMSL). Проте для розрахунку горизонтальної дальності до цілі критично важливо оперувати відносною висотою над рівнем підстилаючої поверхні (AGL).

Для забезпечення найвищої точності розрахунків у розроблюваній системі використовуватиметься лазерний далекомір (за його наявності), який вимірює пряму дистанцію до землі. Оскільки під час польотних маневрів БПЛА може відхилятися від ідеально горизонтального положення, пряма дистанція D , виміряна жорстко закріпленим на корпусі далекоміром, є похилою відстанню. Тому істинна вертикальна висота h обчислюється шляхом тригонометричної корекції на основі поточних кутів крену (φ) та тангажу (θ) самого апарата, отриманих з польотного контролера:

$$h = D \times \cos(\varphi) \times \cos(\theta) \quad (2.4)$$

У разі відсутності лазерного далекоміра або виходу його за межі робочої дальності, алгоритм передбачає автоматичний перехід на резервний метод: розрахунок алгебраїчної різниці між абсолютною барометричною висотою БПЛА та висотою рельєфу в точці знаходження цілі. Дані про рельєф можуть підвантажуватися на наземній станції з глобальних цифрових моделей (Digital

Elevation Model — DEM). Однак все ж рекомендовано використовувати виключно далекомір для точного визначення польотної висоти.

Розрахунок кутів просторової орієнтації (α та β)

Більшість сучасних розвідувальних безпілотних комплексів оснащені трьохосьовими гіростабілізованими підвісами. Апаратна частина підвісу автоматично компенсує вплив кутів крену та тангажу самого літального апарата.

Завдяки цьому визначення абсолютного кута нахилу оптичної осі α значно спрощується: він дорівнює куту нахилу самого підвісу відносно лінії горизонту, що транслюється у стандартному пакеті телеметрії (наприклад, за протоколом MAVLink).

Абсолютний азимут напрямку камери β формується як суперпозиція курсового кута самого БПЛА (Yaw БПЛА) та відносного кута повороту підвісу камери по осі панорамування (Yaw підвісу):

$$\beta = \psi_{UAV} + \psi_{gimbal} \quad (2.5)$$

де ψ_{UAV} — магнітний або істинний курс безпілотника, ψ_{gimbal} — кут повороту камери відносно поздовжньої осі дрона.

Нівелювання впливу оптичних параметрів камери

Відповідно до обраної парадигми наведення по центру (lock-on), розрахунок координат ініціюється виключно в момент, коли ціль утримується в геометричному центрі кадру. Таке інженерне рішення дозволяє повністю виключити з розрахункового алгоритму матрицю внутрішніх оптичних параметрів камери.

Зокрема, математична модель не потребує інформації про фокусну відстань об'єктива, фізичні розміри CMOS-сенсора, розмір окремого пікселя та коефіцієнти радіального чи тангенціального викривлення лінзи. Це значно

знижує обчислювальне навантаження на апаратну частину наземної станції. Найважливішою перевагою такого підходу є забезпечення апаратної незалежності програмного комплексу: розроблена система здатна ефективно функціонувати з відеопотоком від будь-якого типу камери без необхідності проведення її попереднього складного калібрування.

2.3 Алгоритм синхронізації відеопотоку з даними телеметрії за часовими мітками

Критичною умовою забезпечення високої точності розрахунку географічних координат цілі є просторово-часова синхронізація кадрів відеопотоку, на яких нейронна мережа виявляє об'єкт, із відповідним пакетом телеметричних даних (поточними координатами БПЛА, істинною висотою та кутами орієнтації камери). Будь-яка розбіжність у часі між зображенням та метаданими призводить до того, що розрахунковий алгоритм використовує неактуальні просторові параметри.

Враховуючи динаміку польоту (швидкість сучасних розвідувальних крил або мультироторних систем може сягати 15–25 м/с, а кутова швидкість обертання підвісу — десятків градусів за секунду), розсинхронізація навіть у 200–300 мілісекунд здатна згенерувати похибку геолокації на поверхні землі, що вимірюється десятками метрів. Така похибка є критичною та унеможлиблює ефективне цілевказання.

В умовах сучасного театру воєнних дій на стабільність каналів керування та передачі даних суттєво впливає активна робота засобів радіоелектронної боротьби (РЕБ). Інтеграція на борт БПЛА терміналів супутникового зв'язку (зокрема, систем класу Starlink) дозволяє частково нівелювати вплив РЕБ та забезпечити стійкий широкосмуговий канал для трансляції. Однак такий тип маршрутизації вносить специфічні мережеві перешкоди. До них належать нерівномірні, асиметричні та динамічно змінні

затримки (jitter), спричинені перемиканням між супутниками, фазованими антенними решітками та процесами перепакування даних.

Через різну пропускну здатність каналів та багаторівневі процеси буферизації, важкий кадр відео (наприклад, у роздільній здатності 1080p) і легкий текстовий пакет телеметрії, які були згенеровані на борту БПЛА в одну й ту саму мілісекунду, надходять на наземну станцію оператора з різним часовим зміщенням. Тому класичний розрахунок координат на основі часу отримання даних на землі (Time of Arrival) є концептуально хибним.

Для розв'язання цієї проблеми у розроблюваній системі застосовується апаратно-програмний алгоритм жорсткої синхронізації на основі абсолютних часових міток (timestamps), що формуються безпосередньо на борту БПЛА:

1. Апаратна синхронізація бортового часу: Бортовий мікрокомп'ютер, що відповідає за обробку відео, та польотний контролер синхронізують свої внутрішні годинники за протоколом NTP (Network Time Protocol) або за допомогою апаратного сигналу PPS (Pulse Per Second) від модуля супутникової навігації. У разі супресії GPS-сигналу засобами РЕБ відлік ведеться від високоточного бортового годинника реального часу (RTC).
2. Маркування потоків (Timestamping): Кожен відеокадр на етапі кодування інкапсулює у свою структуру (наприклад, через повідомлення SEI — Supplemental Enhancement Information у кодеках H.264/H.265) точну мітку часу t_v . Паралельно кожен телеметричний пакет (MAVLink) отримує власну мітку t_m .
3. Кільцева буферизація на наземній станції: Після отримання даних по радіоканалу програмне забезпечення наземної станції не передає їх одразу на розрахунковий модуль. Телеметричні дані акумулюються у динамічному кільцевому буфері (Ring Buffer) із заданою глибиною зберігання (наприклад, за останні 2–3 секунди).

4. Зіставлення та математична інтерполяція: Коли згорткова нейронна мережа (YOLO) завершує інференс та ідентифікує ціль на кадрі з часовою міткою t_v , алгоритм виконує пошук найближчих «сусідніх» пакетів телеметрії у буфері. Знаходяться два пакети з мітками t_{m1} та t_{m2} , які задовольняють умову: $t_{m1} \leq t_v \leq t_{m2}$.

Оскільки частота кадрів відеопотоку (наприклад, 30 FPS) зазвичай є асинхронною щодо частоти оновлення телеметрії (наприклад, 10–20 Гц), точне значення необхідного телеметричного параметра (наприклад, кута тангажу α) для моменту часу t_v розраховується за допомогою лінійної інтерполяції за формулою:

$$\alpha(t_v) = \alpha(t_{m1}) + (\alpha(t_{m2}) - \alpha(t_{m1})) \times \frac{t_v - t_{m1}}{t_{m2} - t_{m1}} \quad (2.6)$$

Аналогічний інтерполяційний розрахунок застосовується для кута азимута β , висоти h та географічних координат самого безпілотнока.

Застосування запропонованого алгоритму гарантує, що базова математична модель геолокації оперуватиме виключно тими просторовими параметрами, які відповідають фізичному стану БПЛА у точну мілісекунду експозиції кадру. Це забезпечує максимальну точність локалізації малорозмірних цілей та робить систему стійкою до деградації каналів зв'язку та мережевих затримок.

Висновки до розділу

У другому розділі розроблено та теоретично обґрунтовано комплекс математичних моделей і алгоритмів, необхідних для точного визначення географічних координат малорозмірних наземних цілей з борту БПЛА.

За результатами проведеного дослідження зроблено наступні висновки:

1. Оптимізація математичної моделі геолокації. Застосування інженерної парадигми прямого візування («захоплення цілі» в центрі кадру)

дозволило значно спростити алгоритм розрахунку координат, звівши його до методів просторової тригонометрії. Доведено, що такий підхід повністю нівелює вплив внутрішніх оптичних параметрів камери (фокусна відстань, розмір матриці, радіальна дисторсія) та кута крену БПЛА. Це не лише знижує обчислювальне навантаження на систему, але й забезпечує її повну апаратну незалежність від типу застосованого оптико-електронного модуля.

2. Комплексування телеметричних даних. Обґрунтовано методику визначення істинної відносної висоти польоту за допомогою бортового лазерного далекоміра з обов'язковою тригонометричною корекцією на поточні кути нахилу дрона. Також формалізовано процес розрахунку абсолютного азимута та тангажу лінії візування шляхом суперпозиції кутів орієнтації самого безпілотної та його гіростабілізованого підвісу.
3. Забезпечення стійкості в умовах впливу РЕБ. Вирішено критичну проблему просторово-часової розсинхронізації даних, яка виникає через асиметричні мережеві затримки при використанні супутникових систем зв'язку (наприклад, Starlink) та під час активної протидії засобів радіоелектронної боротьби. Розроблено апаратно-програмний алгоритм жорсткої синхронізації на основі вбудовування абсолютних часових міток безпосередньо у відеопотік та телеметрію. Використання кільцевого буфера та математичної інтерполяції гарантує, що розрахунок координат здійснюється виключно за актуальними просторовими параметрами БПЛА на точну мілісекунду фіксації цілі.

Отримані математичні моделі та алгоритмічні рішення формують надійний теоретичний фундамент розроблюваної системи. Наступним етапом дослідження є практичне проектування гібридної програмно-апаратної архітектури комплексу, підготовка спеціалізованого датасету та навчання нейронної мережі для розпізнавання цілей, що буде детально розглянуто у третьому розділі.

РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Базова апаратна оснастка БПЛА (Бортовий сегмент)

Бортовий сегмент у запропонованій гібридній архітектурі виконує виключно функції сенсорної платформи збору первинних даних. Усі ресурсомісткі обчислення (інференс нейронної мережі та геолокаційні розрахунки) винесені на наземну станцію. Такий підхід дозволяє мінімізувати енергоспоживання, масу та тепловиділення на борту, що є критичним для збереження льотних характеристик БПЛА.

Функціонально бортовий сегмент включає такі ключові підсистеми: систему позиціонування, систему візування, лазерні системи вимірювання дальності та висоти, систему передачі даних та систему керування польотом. Також визначальним є вибір платформи-носія, здатної забезпечити необхідну вантажопідйомність.

Система позиціонування

Система позиціонування забезпечує визначення географічних координат БПЛА, висоти польоту, швидкості та курсу. Отримані дані використовуються для навігації, стабілізації, виконання автоматичних режимів польоту, а також для прив'язки сенсорної інформації до глобальної системи координат.

У сучасних БПЛА застосовуються GNSS-модулі, що підтримують одночасну роботу з кількома супутниковими системами (GPS, GLONASS, Galileo, BeiDou). Це підвищує точність, стійкість до завад та зменшує час ініціалізації.

Залежно від вимог до точності, модулі позиціонування поділяються на:

- стандартні (точність 1–3 м);
- професійні (0,5–1 м);
- RTK-модулі (1–3 см).

Для задач геолокації об'єктів доцільним є використання професійних або RTK-рішень, оскільки похибка позиціонування БПЛА прямо впливає на точність визначення координат цілі.

Критичними характеристиками є:

- частота оновлення (не менше 10 Гц);
- підтримка мультиконстеляційності;
- час холодного старту;
- тип антени (активна/пасивна);
- наявність магнітометра;
- інтерфейси (UART, I2C).

Типові апаратні рішення:

- u-blox ZED-F9P — високоточний RTK-модуль;
- Here3 GNSS Module — інтегрований модуль з компасом;
- інтегровані системи у БПЛА виробництва Ukrspesystems та DeVIRo.

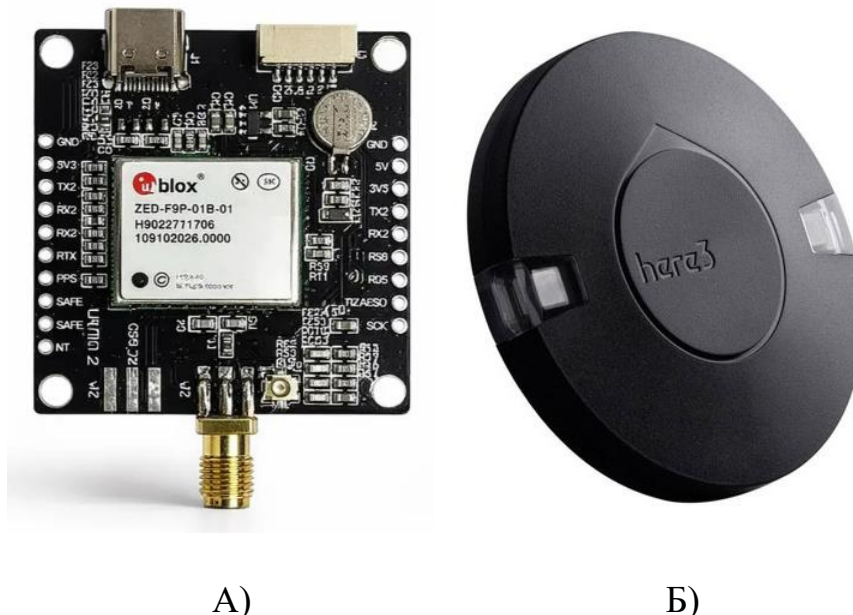


Рисунок 3.1 — Приклади апаратних рішень для геолокації (А - ZED-F9P, Б - Here3 GNSS Module)

Монтаж GNSS-модуля повинен виконуватися з урахуванням електромагнітної сумісності: на віддаленні від джерел завад, з підняттям над корпусом БПЛА та використанням екранованих з'єднань.

Точність GNSS-модуля визначає нижню межу похибки всієї системи геолокації, тому використання професійних або RTK-рішень є технічно доцільним для задач виявлення та прив'язки цілей.

Система візування (оптико-електронний модуль)

Система візування формує відеопотік для алгоритмів комп'ютерного зору.

Склад:

- денна камера;
- тепловізор (опціонально);
- гіростабілізований підвіс.

Критичні параметри:

- оптичний zoom;
- роздільна здатність Full HD (1920×1080);
- стабілізація по трьох осях;
- телеметрія кутів підвісу.

Типові рішення:

- RunCam Split 4 — компактна камера з підтримкою Full HD, низькою затримкою та малою масою;
- Caddx Ratel 2 — FPV-камера з високою світлочутливістю, придатна для роботи в умовах низької освітленості;
- Foxeer Toothless 2 — камера з широким динамічним діапазоном (WDR), що покращує якість зображення в складних сценах;

- бюджетні цифрові модулі на базі Raspberry Pi Camera Module v2 або аналогічних сенсорів Sony.



А)

Б)

Рисунок 3.2 — Приклади апаратних рішень для візування цілей (А - RunCam Split 4, Б - Caddx Ratel 2)

Використання Full HD є оптимальним компромісом між інформативністю та бітрейтом, тоді як ключовим фактором точності є стабілізація та можливість утримання цілі в центрі кадру.

Система визначення висоти

Визначення істинної висоти польоту БПЛА відносно підстилаючої поверхні (AGL — Above Ground Level) є критично важливим параметром для задач геолокації об'єктів. Помилка у визначенні висоти безпосередньо масштабує похибку координат цілі, особливо при великих кутах нахилу камери.

Основним методом визначення висоти у запропонованій системі є використання лазерного далекоміра, орієнтованого у напрямку підстилаючої поверхні.

Характеристики:

- діапазон вимірювання: від кількох метрів до 200–500 м і більше;

- точність: від сантиметрів до $\sim \pm 1$ м (залежно від дистанції);
- частота оновлення: 10 – 100 Гц і більше;
- кут розбіжності променя: визначає точність «попадання» у поверхню;
- стійкість до зовнішніх факторів (сонячне світло, туман, пил).

Лазерні далекоміри забезпечують стабільне вимірювання висоти при польотах на різних висотах і при маневруванні, що робить їх значно точнішими за барометричні методи.

Типові рішення:

- LightWare GRF-500 — лазерний далекомір з дальністю до 500 м, оптимізований для БПЛА;
- Venewake TFA1500 — професійне рішення з дальністю до 1500 м;
- інтегровані модулі у DJI Zenmuse H20T.

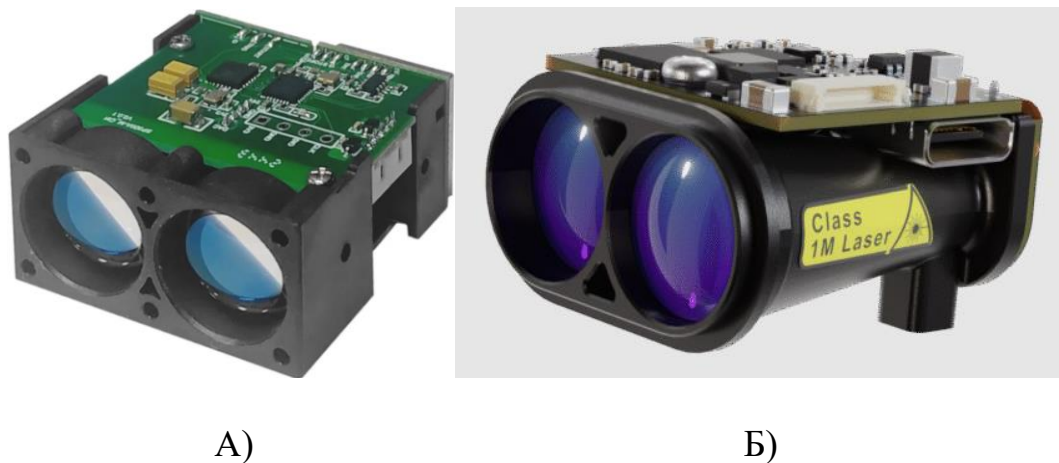


Рисунок 3.3 — Приклади апаратних рішень далекомірів (А - Venewake TF1500, Б - LightWare GRF-500)

Використання далекомірів із дальністю не менше 500 м є мінімальною необхідною умовою для забезпечення коректної геометрії задачі при роботі з БПЛА. Компактні TOF-сенсори забезпечують оптимальний баланс між масою та точністю, тоді як далекоміри розширеного діапазону дозволяють підвищити робочі дистанції ціною ускладнення системи.

Система передачі даних

Система передачі даних забезпечує безперервну трансляцію відеопотоку та телеметричної інформації з борту БПЛА на наземну станцію. У межах запропонованої архітектури дана підсистема є критичною, оскільки саме через неї передається вся інформація, необхідна для роботи алгоритмів комп'ютерного зору та подальшої геолокації об'єктів.

На відміну від класичних автономних БПЛА, у яких обчислення виконуються на борту, у даному випадку канал зв'язку стає ключовим елементом системи, що визначає її ефективність.

1) Структура системи

Система передачі даних включає кілька логічно розділених каналів:

- відеоканал — передача відеопотоку з камери (основне джерело даних для комп'ютерного зору);
- телеметричний канал — передача параметрів польоту (координати, кути орієнтації, швидкість тощо);
- службовий канал управління — передача команд управління БПЛА (опціонально);
- IP-канал (мережевий) — використовується при наявності ретрансляції (наприклад, через супутниковий зв'язок).

2) Вимоги до системи

З урахуванням особливостей задачі, до системи передачі даних висуваються такі вимоги:

- пропускна здатність: не менше 4–8 Мбіт/с для стабільного Full HD відео;
- затримка (latency): мінімально можлива, бажано до 200–300 мс;
- стійкість до завад (РЕБ): використання адаптивних протоколів, частотного перестроювання, шифрування;
- надійність зв'язку: підтримка втрат пакетів та відновлення потоку;

- синхронізація даних: підтримка часових міток (timestamps).

Особливістю запропонованої системи є те, що обробка даних виконується не за часом їх отримання, а за абсолютними часовими мітками, які формуються на борту.

3) Синхронізація відео та телеметрії

Для забезпечення коректної геолокації необхідна точна синхронізація відеопотоку та телеметричних даних. У системі використовується підхід із вбудованими часовими мітками:

- відео: timestamps вбудовуються у потік (SEI — Supplemental Enhancement Information);
- телеметрія: timestamps передаються у повідомленнях протоколу MAVLink.

На наземній станції реалізується буферизація потоків із подальшою інтерполяцією телеметрії відповідно до часових міток кадрів відео. Це дозволяє компенсувати нестабільні затримки каналу зв'язку (джиттер), характерні для мереж типу IP або супутникових каналів.

Такий підхід забезпечує високу точність узгодження даних навіть в умовах значних мережових затримок.

4) Технології передачі

У практичних реалізаціях застосовуються такі технології:

- цифрові HD-лінки (OFDM) — основний варіант для передачі відео;
- Wi-Fi/802.11 модифіковані рішення — для кастомних систем;
- LTE/4G/5G — як додатковий канал;
- супутниковий зв'язок (Starlink) — для роботи на великих відстанях через ретрансляцію.

Типові рішення:

- Herelink HD Video System — інтегрований відео- та телеметричний канал;
- DJI O3 Air Unit — цифровий відеолінк;
- OpenHD — відкриті рішення на базі Wi-Fi;
- рішення з використанням Starlink як ретранслятора.

5) Особливості реалізації в умовах РЕБ

Умови радіоелектронної протидії вимагають:

- використання цифрових протоколів із корекцією помилок;
- динамічного вибору частот;
- можливості швидкого перемикання каналів;
- буферизації та повторної передачі пакетів.

Застосування IP-орієнтованої архітектури (через Starlink або інші канали) дозволяє підвищити дальність, але збільшує затримку, що компенсується використанням timestamp-синхронізації.

Система передачі даних є критичним елементом гібридної архітектури, оскільки визначає можливість винесення обчислень на наземну станцію. Використання HD/Full HD відео, адаптивних каналів зв'язку та обов'язкової синхронізації через часові мітки забезпечує коректну роботу системи навіть в умовах нестабільного зв'язку.

Система керування польотом

Система керування польотом реалізується у складі базової платформи БПЛА та постачається виробником як інтегрований компонент. Вона включає польотний контролер (автопілот), набір інерціальних сенсорів та програмне забезпечення, необхідне для стабілізації та навігації.

У межах даної роботи система керування польотом не підлягає модифікації або розробці, а використовується як готовий функціональний модуль.

Функціональне призначення

На систему керування польотом покладаються такі задачі:

- стабілізація БПЛА у просторі (roll, pitch, yaw);
- визначення параметрів руху на основі IMU та GNSS;
- реалізація автоматичних режимів польоту;
- керування виконавчими механізмами (двигуни, серводи);
- формування телеметричного потоку.

Система працює у реальному часі з високими частотами оновлення, що забезпечує стабільність польоту навіть у складних умовах.

Система керування польотом виступає джерелом телеметричних даних для наземної станції та забезпечує узгодження роботи всіх бортових підсистем:

- GNSS-модуля;
- лазерного далекоміра;
- підвісу камери;
- системи передачі даних.

Система керування польотом розглядається як готовий інтегрований компонент БПЛА, який забезпечує стабілізацію та формування телеметрії. Використання стандартних протоколів обміну (MAVLink) дозволяє ефективно інтегрувати її у загальну систему без необхідності модифікації бортового програмного забезпечення.

Вимоги до платформи-носія

Платформа-носіїв БПЛА визначає можливість інтеграції всіх бортових підсистем та безпосередньо впливає на ефективність роботи системи

виявлення та геолокації об'єктів. У межах даної роботи платформа розглядається як базовий елемент, що забезпечує підйом, енергоживлення та стабілізацію сенсорного навантаження.

1) Вантажопідйомність

Ключовою вимогою є забезпечення підйому корисного навантаження масою не менше 5–8 кг (з можливістю розширення до 10 кг).

До складу навантаження входять:

- оптико-електронний модуль (камера + підвіс);
- лазерний далекомір;
- GNSS-модуль;
- система передачі даних;
- допоміжні модулі та кабельна інфраструктура.

Недостатня вантажопідйомність призводить до компромісів у виборі сенсорів або зменшення часу польоту.

2) Тривалість польоту

Для забезпечення практичної цінності системи необхідна тривалість польоту не менше 30–60 хвилин.

Цей параметр залежить від:

- маси корисного навантаження;
- ємності акумуляторів;
- типу силової установки;
- аеродинамічних характеристик платформи.

Збільшення тривалості польоту дозволяє:

- розширити зону спостереження;
- підвищити ймовірність виявлення цілей;
- зменшити кількість вильотів.

3) Стабільність платформи

Стабільність польоту є критичною для роботи системи комп'ютерного зору.

Вона визначається:

- ефективністю системи керування польотом;
- конструкцією рами;
- рівнем вібрацій;
- якістю балансування пропелерів.

Недостатня стабільність призводить до:

- розмиття зображення;
- зниження точності детекції;
- ускладнення утримання цілі у центрі кадру.

4) Надійність та ремонтпридатність

З урахуванням умов експлуатації платформа повинна відповідати таким вимогам:

- стійкість до механічних навантажень;
- можливість швидкого обслуговування;
- модульність конструкції;
- доступність запасних частин.

Ці фактори є критичними для забезпечення безперервної експлуатації системи.

5) Типові платформи

У практичних реалізаціях можуть використовуватися такі БПЛА:

- Freefly Alta 8;
- PD-2 UAV;

Вибір конкретної платформи залежить від бюджету, доступності та вимог до місії.



Рисунок 3.4 — Приклади апаратних рішень далекомірів (А - PD-2 UAV, Б - Freely Alta 8)

Платформа-носії визначає можливість реалізації всієї системи без необхідності компромісів у виборі сенсорів. Забезпечення вантажопідйомності на рівні 5–8 кг, достатньої тривалості польоту та високої стабільності є обов’язковими умовами ефективної роботи системи комп’ютерного зору та геолокації.

3.2 Архітектура наземної системи керування та збору даних (Наземний сегмент)

Наземний сегмент розроблюваної системи є ключовим обчислювальним ядром комплексу, де зосереджені основні потужності для виконання задач машинного зору та прецизійної геолокації. На відміну від бортового сегмента, який виконує роль сенсорної платформи, наземна станція керування (Ground Control Station — GCS) забезпечує повний цикл обробки інформації: від прийому сирого відеопотоку до візуалізації верифікованих цілей на інтерактивній карті.

Згідно з інженерною концепцією, архітектура наземного сегмента включає такі функціональні модулі:

Модуль прийому та демультимплексування даних

Ця підсистема відповідає за стабільне отримання гетерогенних потоків інформації через канали зв'язку. На цьому етапі відбувається розділення вхідного трафіку на дві основні гілки:

- **Відеоінформація:** декомпресія відеопотоку високої роздільної здатності для подальшої передачі на аналіз алгоритмами машинного зору.
- **Телеметричний пакет:** вилучення координат дрона ($X1, Y1$), поточної висоти (h), дистанції від далекоміра (D) та кутових параметрів орієнтації камери.

Обчислювальний модуль машинного зору (Inference Engine)

Це найбільш ресурсомістка частина системи, реалізована на базі високопродуктивних графічних процесорів (GPU). Модуль здійснює безперервний аналіз відеоряду за допомогою архітектури YOLO.

Застосування технологій машинного зору на наземній станції дозволяє використовувати повнорозмірні нейронні мережі з високою роздільною здатністю вхідного зображення. Це забезпечує надійне виявлення малорозмірних об'єктів у реальному часі, що було б неможливим при використанні обмежених потужностей бортових мікрокомп'ютерів.

Модуль математичної обробки та геолокації

Підсистема, що інтегрує результати роботи машинного зору з даними телеметрії для розрахунку координат цілі. Вона реалізує математичну модель, яка на основі параметрів h, D, α, β миттєво обчислює географічне положення об'єкта. Також модуль здійснює часову синхронізацію, що нівелює затримки в каналах зв'язку.

Геоінформаційна система (ГІС) та інтерфейс оператора

Візуальний компонент системи, призначений для відображення ситуаційної обстановки.

- Картографія: автоматичне нанесення на карту маркерів положення БПЛА та виявлених цілей у режимі реального часу.
- Семантика об'єкта: можливість виведення додаткових даних при взаємодії з маркером, включаючи розраховані координати та фрагмент відео для візуальної верифікації об'єкта оператором.

Система архівації та бази даних

Для забезпечення аналізу результатів передбачено базу даних, у якій зберігаються:

- Відеозапис польоту.
- Логи координат та висот безпілотної літака.
- Реєстр виявлених цілей із розрахованими географічними координатами.

Вимоги до апаратного забезпечення:

Для стабільного функціонування системи машинного зору наземна станція повинна бути реалізована на базі обчислювальної платформи з дискретною відеокартою та багатоядерним процесором, що забезпечить паралельну обробку відео та математичних алгоритмів без затримок.

Розподіл функцій між бортом та наземною станцією дозволяє створити гнучку та апаратно-незалежну систему, яка мінімізує вплив людського фактора та забезпечує високу точність отримання розвідувальних даних.

3.3 Підготовка датасету та навчання нейронної мережі YOLO

Якість, обсяг та репрезентативність вхідних даних відіграють ключову роль у забезпеченні ефективності систем, що базуються на згорткових нейронних мережах. Особливо це актуально для тактичної аеророзвідки, де

наземні цілі часто приховані від прямого огляду, частково закриті елементами рельєфу або демонструють нетипові візуальні характеристики, як-от присутність десанту на бронетехніці. Виходячи з таких викликів, до процесу підготовки даних було поставлено високі вимоги. Для створення моделі розпізнавання багатоцільового тягача МТ-ЛБ застосовано комплексний підхід, що охоплює збір, маркування та обробку даних. При цьому здійснювалося порівняння двох різних підходів у комп'ютерному зорі: класичної детекції об'єктів (Object Detection) та інстанс-сегментації (Instance Segmentation).

3.3.1 Формування навчальної вибірки (набору даних)

Специфіка навчання нейромереж для потреб БПЛА полягає в тому, що класичні фотографії військової техніки (зроблені з рівня землі) є малоінформативними. Моделі необхідні ракурси згори під різними кутами тангажу камери. Для досягнення максимальної здатності мережі до узагальнення (generalization), навчальну вибірку було сформовано шляхом агрегації даних з двох незалежних джерел:

1. Базовий набір (Відкриті репозиторії): Основу датасету склали зображення з платформи машинного навчання Roboflow Universe, зокрема з набору даних «*Russian Military Vehicles*»[3]. Це забезпечило модель достатньою кількістю базових ракурсів техніки в різних погодних умовах та на різних типах місцевості.
2. Специфічний набір (Проект Oryx): Для підвищення стійкості алгоритму до перешкод та умов реальних бойових дій, було проведено цільовий збір фото- та відеоматеріалів з аналітичного ресурсу *Oryx (Project Spioenkor)*[4]. Долучення кадрів знищеної, пошкодженої техніки, а також машин із нестандартним польовим маскуванням (сітки, гілля, деформуюче фарбування) дозволило значно розширити простір ознак,

які виділяє нейромережа, унеможлиблюючи її перенавчання (overfitting) на «ідеальні» зображення полігонних зразків.

3.3.2 Методика розмітки у середовищі Label Studio

Процес анотування зібраних даних, тобто перетворення необроблених пікселів зображення у математичні координати, придатні для аналізу нейронною мережею, здійснювався за допомогою спеціалізованого програмного забезпечення Label Studio. У ході цього дослідження був проведений порівняльний експеримент, у рамках якого датасет розмічався двома різними підходами.

- Метод Bounding Box (обмежувальні рамки) використовує класичні прямокутники для виділення об'єктів. Анотаційний файл зберігає координати у форматі [клас, x_center, y_center, width, height]. Такий підхід є швидким у реалізації, але практика показує деякі його недоліки [6]. Наприклад, з висоти польоту БПЛА видовжений силует МТ-ЛБ часто розташовується під кутом до осей кадру. Через це прямокутна рамка охоплює значну частину фонового шуму, такого як земля чи трава — близько 40-50%. Це може призвести до того, що нейромережа помилково врахує ці елементи як характеристики самого об'єкта.
- Метод полігональної сегментації (Instance Segmentation) передбачав створення точного попиксельного окреслення складних контурів МТ-ЛБ за допомогою полігональних координат [клас, x1, y1, ... xn, yn] [7]. Хоч ручна розмітка є досить ресурсозатратною, цей підхід все ж ефективніше ізолює об'єкт від навколишнього середовища. Це сприяє тому, що алгоритм може фокусуватися виключно на конструктивних деталях об'єкта, таких як катки, гусениці та форма корпусу.



А)

Б)

Рисунок 3.5 — Методи Bounding Box та Instance Segmentation

3.3.3 Процес навчання нейронних мереж

Для емпіричного підтвердження гіпотези щодо переваги попиксельного розпізнавання над класичним виділенням об'єктів рамками, було проведено двоетапне навчання. Тренування обох моделей здійснювалося в ідентичному програмно-апаратному середовищі (мовою Python 3.13 з використанням фреймворку PyTorch) на апаратному прискорювачі NVIDIA GeForce RTX 3050 Laptop GPU (4GB VRAM). Роздільна здатність вхідних зображень (imgsz) становила 640x640 пікселів, розмір батчу (batch) було оптимізовано до 8 для уникнення переповнення відеопам'яті.

Етап 1: Базова детекція (Bounding Box)

У якості baseline-рішення було обрано глибоку архітектуру YOLOv8l (Large). Навчання проводилося протягом 60 епох. Під час тренування алгоритм стохастичного градієнтного спуску оптимізував три цільові функції втрат: `box_loss` (похибка передбачення координат рамки), `cls_loss` (похибка класифікації) та `dfl_loss` (похибка фокального розподілу меж). Результати навчання наведено в Рис. 3.6.

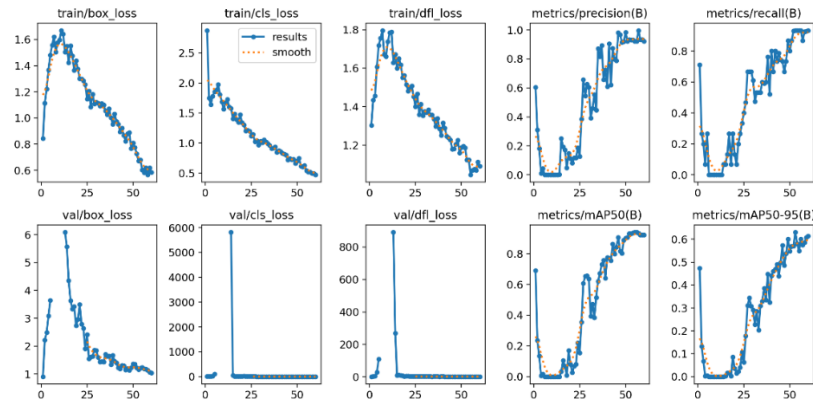


Рисунок 3.6 — Результати навчання YOLOv8l методом Bounding Box

Етап 2: Поглиблене розпізнавання (Instance Segmentation)

Для тестування сегментувального підходу використано сучасну архітектуру YOLO26l-seg. Оскільки сегментаційна модель вирішує значно складніше завдання, яке включає не лише локалізацію об'єктів у просторовій рамці, але й генерацію точної бінарної маски об'єкта (з мінімізацією додаткової функції втрат `seg_loss`), тривалість навчального циклу була збільшена до 100 епох. Результати навчання наведено в Рис. 3.7.

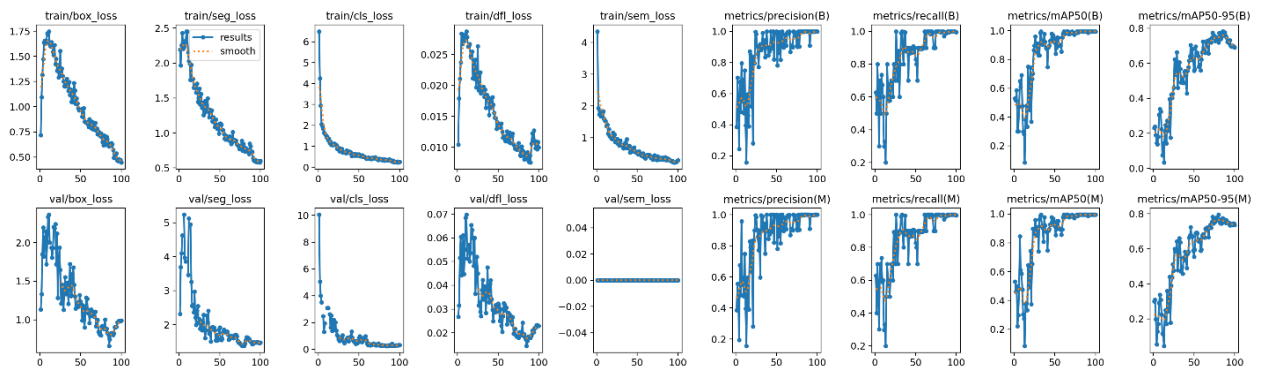


Рисунок 3.7 — Результати навчання YOLOv26l методом Instance Segmentation

Остаточний вибір моделі YOLO26l-seg для ідентифікації техніки ґрунтувався на аналізі продуктивності різних версій архітектури (рис. 3.8). Як видно з бенчмарку, версія «Large» забезпечує оптимальний компроміс: вона знаходиться у точці вигину кривої, де подальше збільшення кількості параметрів (перехід до версії «x») дає незначний приріст точності при

суттєвому зростанні обчислювальних витрат. Це дозволяє реалізувати сегментацію в режимі реального часу на апаратних потужностях наземної станції».

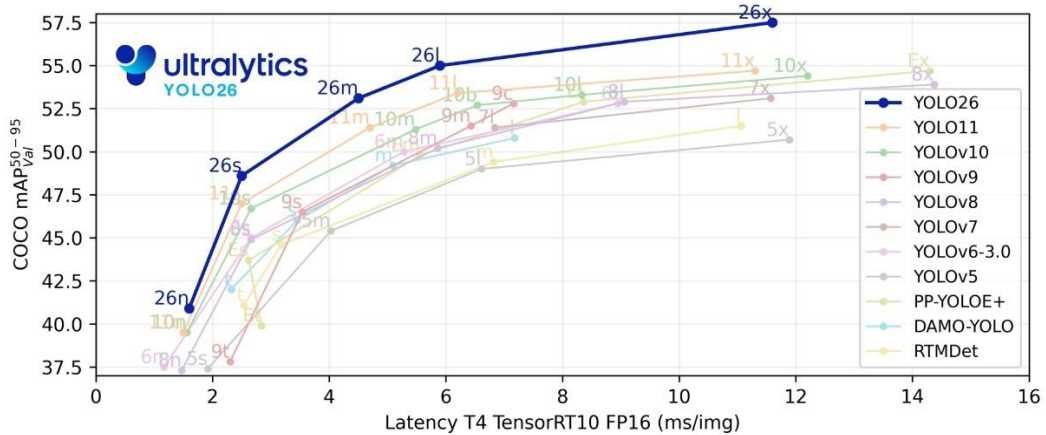


Рисунок 3.8 — Порівняльні характеристики продуктивності архітектур сімейства YOLO

3.3.4 Порівняльний аналіз результатів навчання

Для кількісної оцінки ефективності навчених моделей було проведено детальний аналіз динаміки зміни їхніх цільових метрик (Precision, Recall, mAP) на валідаційній вибірці. Найвищі зафіксовані показники з логів навчання зведено у порівняльну таблицю 3.1.

Таблиця 3.1 — Порівняння метрик ефективності базової та сегментаційної моделей YOLO

Метрика якості (Metrics)	YOLOv8l (Bounding Box)	YOLO26l-seg (Instance Segmentation)
Кількість епох	60	100
Precision (Точність)	0.920	0.988
Recall (Повнота)	0.933	1.000
mAP@50	0.939	0.995
mAP@50-95	0.549	0.636

Обмеження традиційного підходу до детекції: Модель YOLOv8l показала високу здатність виявлення об'єктів (Recall 93.3%) та досить хорошу середню точність (mAP@50 близько 94%). Проте основною проблемою став показник жорсткої метрики mAP@50-95(B), який передбачає ідеальну відповідність передбаченої рамки реальним габаритам техніки на основі коефіцієнта перекриття області (IoU). Значення цієї метрики залишається на рівні 0.549, що свідчить про вплив «фонового шуму»: недосконалість контуру призводить до невпевненої масштабної оцінки рамки навколо маскованих об'єктів.

Переваги сегментного підходу: Перехід до моделі YOLO26l-seg забезпечив суттєве покращення всіх ключових параметрів. Завдяки сегментації, яка ізолює пікселі об'єкта від фону через полігональну маску, метрика повноти виявлення (Recall) досягла абсолютного значення — 1.000, що засвідчило відсутність пропущених цілей класу МТ-ЛБ на тестовій вибірці. Однак при подальших тестах на ідентифікацію об'єктів алгоритм зрідка пропускав техніку, що вказує на необхідність додаткової оптимізації.

Точність просторової локалізації: Середня точність накладання згенерованої полігональної маски mAP@50 досягла вражаючих 99.5%. Разом з цим, жорсткий критерій mAP@50-95 зріс на 15.8% (з 0.549 до 0.636), що свідчить про високу геометричну точність виділення силуету об'єкта, незалежно від кута його повороту відносно камери БПЛА.

Аналіз обмежень та критичних випадків роботи моделі

Окрім успішних прикладів детекції, було проведено аналіз випадків, де модель інстанс-сегментації YOLO26l-seg демонструвала нестійку роботу. Один із таких прикладів наведено на рисунку 3.9.

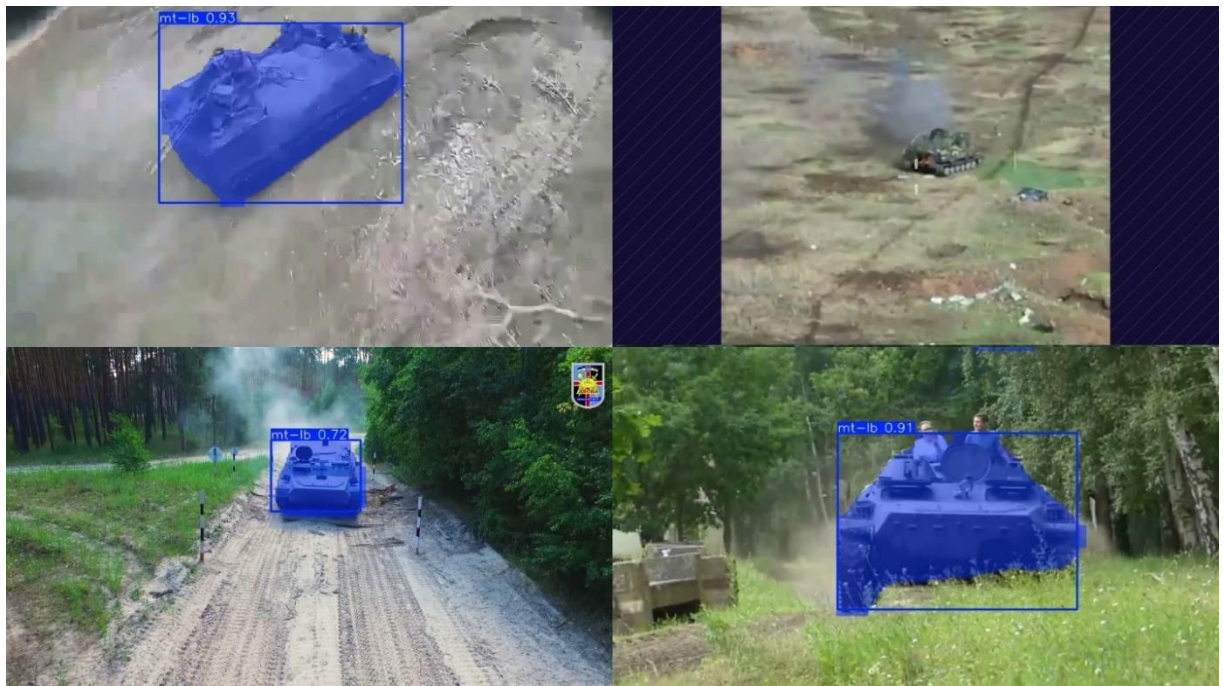


Рисунок 3.9 — Приклад роботи машинного зору

На даному кадрі видно, що модель не змогла впевнено сегментувати МТ-ЛБ. Це обумовлено сукупністю негативних факторів:

1. Низький візуальний контраст та роздільна здатність: Об'єкт знаходиться на значній дистанції, займає малу кількість пікселів та має низьку роздільну здатність. Це призводить до злиття текстури броні з фоном рельєфу (багнюка, трава), що ускладнює попіксельне виділення силуету.
2. Складність ракурсу: Модель, ймовірно, недостатньо навчена на подібні ракурси при такому масштабі, що призвело до невпевненої генерації маски.

Даний випадок чітко підтверджує, що для забезпечення надійної роботи автоматизованої системи в реальних бойових умовах необхідно:

- Значно розширити датасет, додавши більше зображень малорозмірних і низькодеталізованих цілей, а також цілей, знятих у складних умовах, таких як відблиски, туман чи сутінки.
- Модернізувати апаратне забезпечення БПЛА: для ефективної роботи розроблених алгоритмів зйомка повинна виконуватися камерами з

роздільною здатністю не менше HD (1280x720) і підтримкою високоякісного оптичного зуму. Це забезпечить можливість оператору наближати об'єкт, що сприятиме його точнішій сегментації.

3.4 Розробка програмного забезпечення з інтеграцією Google Maps API

Розроблена архітектура програмного модуля виконує функцію сполучної ланки (Backend-ядра) між сирими даними відеопотоку, математичним апаратом нейромережі та кінцевим інтерфейсом картографічної візуалізації. Головне завдання розробленого програмного забезпечення полягає у безперервному парсингу даних, детекції об'єктів, розрахунку їхніх абсолютних просторових координат та формуванні структурованого масиву верифікованої інформації, готової для інтеграції з API Google Maps. (Код з подальших розділів відноситься до Лістингу коду з Додадку А)

3.4.1 Структуризація та синхронізація вхідних телеметричних даних

Для забезпечення належного функціонування математичного апарату системи просторової прив'язки даних дані про політ БПЛА імпортуються з CSV-журналів в оптимізовану структуру даних `pandas.DataFrame`. Використання бібліотеки `pandas` дозволяє забезпечити векторний доступ до масивів телеметричних даних із мінімальними затримками при виконанні (обчислювальна складність $O(1)$ для операцій доступу за індексом рядка).

Вхідний структурований потік телеметрії містить шість обов'язкових параметрів для кожного дискретного моменту вимірювання:

1. `timestamp` — часова мітка від початку запису (в секундах), що забезпечує базу для синхронізації з відеорядом;
2. `drone_lat` — поточна географічна широта точки знаходження БПЛА у градусах (система координат WGS 84);

3. `drone_lon` — поточна географічна довгота точки знаходження БПЛА у градусах;
4. `alt` — висота польоту літального апарату над рівнем поверхні (в метрах), що визначає масштаб геометричної проекції;
5. `gimbal_pitch` — кут нахилу оптичної осі камери відносно площини горизонту (в градусах, де від'ємні значення відповідають нахилу вниз);
6. `drone_heading` — поточний курс (кутовий азимут) БПЛА відносно географічного півночі (в градусах від 0° до 360°).

Типова структура набору телеметричних даних, що зчитується з файлу та передається в аналітичне ядро на кожній ітерації головного циклу обробки, наведена у таблиці 3.2.

Таблиця 3.2 — Фрагмент масиву вхідних телеметричних даних БПЛА

timestamp	drone_lat	drone_lon	alt	gimbal_pitch	drone_heading
1.85	48.487327	37.098645	118.4	-46.0	120.1
1.901	48.487327	37.098645	118.4	-46.1	120.2
...

3.4.2 Алгоритм інтерполяції та синхронізації кадрів

Основна проблема інтеграції відеопотоку та телеметричних даних з БПЛА полягає у розбіжності частоти дискретизації: відеопотік має фіксовану частоту кадрів, тоді як дані телеметрії записуються сенсорами з індивідуальним асинхронним інтервалом. Для усунення цієї розсинхронізації був розроблений алгоритм точної часової прив'язки.

Процес синхронізації здійснюється через динамічний розрахунок поточного часу кадру відеоряду з використанням його індексу (`frame_idx`) та вихідної частоти кадрів відео (FPS), що автоматично визначається за допомогою методів бібліотеки OpenCV (`cv2.CAP_PROP_FPS`). Отримана

часова мітка слугує основою для пошуку найближчого відповідного запису в структурі `telemetry_data`. Завдяки цьому забезпечується точне відображення подій на відео у співвідношенні із просторовими параметрами БПЛА до конкретної мілісекунди польоту.

3.4.3 Імплементация нейромережевого трекінгу та апаратна оптимізація

Для виявлення та відстеження цілей інтегровано архітектуру YOLOv26. Програмно використовується метод `model.track`, який не лише виконує попиксельну сегментацію об'єктів у кожному кадрі, але й імплементує алгоритми асоціації даних для присвоєння кожному виявленому об'єкту унікального ідентифікатора (Track ID) [8]. Це дозволяє системі стабільно відстежувати об'єкт у поточному кадрі як той самий екземпляр техніки, що й у попередніх кадрах, навіть у випадку часткового перекриття або зміни ракурсу спостереження.

З метою забезпечення швидкодії системи в режимі реального часу (Real-time) на мобільних обчислювальних станціях (Edge Computing), програмна імплементация інференсу використовує апаратне прискорення. Завдяки параметру `device=0` обчислення виконуються безпосередньо на CUDA-ядрах дискретного графічного процесора (GPU). Додатково застосовано оптимізацію половинної точності (`half = True`), що конвертує ваги нейромережі з формату FP32 у FP16, дозволяючи задіяти спеціалізовані тензорні ядра GPU. Це зменшує споживання відеопам'яті та підвищує фреймрейт обробки без втрати точності розпізнавання.

Для усунення хибних спрацьовувань (False Positives) на етапі інференсу встановлено поріг впевненості (Confidence Threshold) на рівні 60% (`conf = 0.6`), а також програмно відфільтровано класи виявлення (`classes = [0]`), що змушує нейромережу реагувати виключно на цільовий клас об'єктів.

3.4.4 Математична модель геопросторової прив'язки об'єктів

Оскільки API картографічних сервісів оперує абсолютними географічними координатами у системі WGS 84, розроблено алгоритм розрахунку просторового зміщення знайденої цілі відносно точки проекції БПЛА на поверхню Землі. Відстань на площині (d_{ground}) від надиру дрона до цілі обчислюється на основі висоти польоту (alt) та компенсації кута нахилу камери відносно горизонту (gimbal_pitch). Програмна реалізація розрахунку дельти зміщення враховує кривизну поверхні Землі, апроксимовану сферою із екваторіальним радіусом $R = 6378137$ м. Функція обчислення абсолютних GPS-координат цілі мовою Python наведена в коді нижче (весь наведений код в цьому розділі і в подальших, є частинами Додатку А).

```
def calculate_target_gps(uav_lat, uav_lon, alt, pitch, heading):
    R = 6378137
    dist_on_ground = alt * math.tan(math.radians(90 + pitch))
    d_lat = (dist_on_ground * math.cos(math.radians(heading))) / R
    d_lon = (dist_on_ground * math.sin(math.radians(heading))) / (R *
    math.cos(math.radians(uav_lat)))
    return uav_lat + math.degrees(d_lat), uav_lon + math.degrees(d_lon)
```

3.4.5 Алгоритм просторово-часової фільтрації та дедуплікації даних

Під час виконання місії система може багаторазово фіксувати одну й ту саму ціль під різними кутами або тимчасово втрачати її через перешкоди рельєфу. Для забезпечення достовірності даних та уникнення засмічення кінцевої карти надлишковими маркерами, розроблено багатокритеріальний алгоритм верифікації та дедуплікації.

На першому етапі застосовується часова верифікація: розроблена стейт-машина (`tracked_targets`) фіксує час першої появи цілі в кадрі. Об'єкт отримує статус верифікованого (`verified = True`) лише за умови його безперервного відстеження нейромережею протягом $\Delta t \geq 3.0$ секунд. Для відеопотоку з частотою 60 FPS це еквівалентно утриманню цілі протягом 180 кадрів поспіль,

що повністю виключає випадкові спрацьовувань на візуальний шум або короточасні артефакти стиснення відеопотоку. Код наведено нижче.

```
# Логіка фільтрації дублікатів у головному циклі програми
    if (current_time - target['first_seen'] >= 3.0) and not
target['verified']:
        target['verified'] = True

        t_lat, t_lon = calculate_target_gps(
            tele_row['drone_lat'], tele_row['drone_lon'],
            tele_row['alt'], tele_row['gimbal_pitch'],
tele_row['drone_heading']
        )

        is_duplicate = False
        for past_tgt in verified_history:
            dist = haversine_distance(t_lat, t_lon, past_tgt['lat'],
past_tgt['lon'])

            time_diff = current_time - past_tgt['timestamp']
            if dist < 2000 and time_diff < 300:
                is_duplicate = True
                print(f"[ФІЛЬТР] Ігнорую ціль #{track_id}. Вже є на
карті.")
                break

        if not is_duplicate:
            print(f"[УСПИХ] Нова ціль #{track_id} підтверджена!
Відкриваю карту...")
```

На другому етапі виконується просторова фільтрація. Обчислюється геодезична відстань між нововиявленою ціллю та всіма раніше підтвердженими об'єктами з використанням методів сферичної тригонометрії, зокрема формули гаверсинусів, що наведено в коді нижче.

```
def haversine_distance(lat1, lon1, lat2, lon2):
    R = 6371000
    phi1, phi2 = math.radians(lat1), math.radians(lat2)
    d_phi = math.radians(lat2 - lat1)
    d_lambda = math.radians(lon2 - lon1)
    a = math.sin(d_phi/2)**2 + math.cos(phi1)*math.cos(phi2) *
math.sin(d_lambda/2)**2
```

```
return R * (2 * math.atan2(math.sqrt(a), math.sqrt(1-a)))
```

3.5 Реалізація інтерфейсу оператора та механізмів верифікації цілей

В результаті розробки системи оператору/командиру розрахунку передається інформаційна панель для прийняття рішення, що є кінцевим етапом обробки даних. Оскільки система працює в режимі-реального часу - Інтерфейс має забезпечувати відразу сповіщати виявлення мети, надати точні геопросторові координати та включати об'єктивне відеозасвідчення для подальшої верифікації результатів роботи нейромережі людиною.

3.5.1 Буферизація відеоданих та експорт доказів

Для забезпечення можливості візуального підтвердження виявленої техніки (зокрема, МТ-ЛБ), паралельно з аналітичною обробкою координат реалізовано підсистему кешування відеоданих. У момент програмного підтвердження нової цілі (утримання в кадрі ≥ 3.0 с), система автоматично ініціює запис збереженого в оперативній пам'яті масиву кадрів (останні 5 секунд) у відеофайл. Програмну реалізацію даного процесу наведено в коді нижче.

```
def save_video_buffer(frames_deque, filename, video_fps):
    if not frames_deque: return
    h, w, _ = frames_deque[0].shape

    fourcc = cv2.VideoWriter_fourcc(*'avc1')
    out = cv2.VideoWriter(filename, cv2.CAP_MSMF, fourcc,
video_fps, (w, h))
    for f in frames_deque: out.write(f)
    out.release()
```

Використання стандарту стиснення H.264 гарантує подальше безпроблемне відтворення згенерованих .mp4 файлів у браузері без використання сторонніх медіаплеєрів.

3.5.2 Динамічна генерація картографічного інтерфейсу

Основним робочим простором оператора є інтерактивна карта, яка генерується на основі масиву верифікованих цілей `verified_history`. Для програмного формування веб-документа використовується бібліотека `Folium` (Python-обгортка над фреймворком `Leaflet.js`).

Для кожної унікальної цілі створюється маркер, у спливаюче вікно (HTML Popur) якого інкапсулюється тег `<video>` з прив'язкою до експортованого відеодоказу. Фрагмент відпрацювання нейромережі на збереженому відео (з відображенням `Track ID` та рівня впевненості моделі) наведено на Рисунку 3.10.



Рисунок 3.10 — Фрагмент відеодоказу з відображенням результатів сегментації та ідентифікатора цілі

Система автоматично присвоює цілям послідовні ідентифікатори (`Track ID`). Пропуски у нумерації (наприклад, наявність цілі з `ID 3` за відсутності `ID 1` та `ID 2`) свідчать про ефективну роботу алгоритму просторово-часової фільтрації: попередні треки цілей були відхилені як нестабільні або візуальні шуми, оскільки не подолали поріг безперервного утримання у 3 секунди.

3.5.3 Розробка інтерактивної панелі верифікації

Для забезпечення взаємодії оператора з системою, у структуру спливаючого вікна маркера інтегровано панель керування на базі JavaScript. Вона реалізує фронтенд-логіку верифікації та класифікації. Оператор має змогу:

1. Підтвердити ціль: вікно забарвлюється у зелений колір, фіксуючи об'єкт для передачі на засоби ураження.
2. Позначити для дорозвідки: вікно забарвлюється у жовтий колір, позначаючи об'єкт для додаткової дорозвідки.
3. Видалити ціль: маркер повністю і безповоротно стирається з картографічного шару.

Щоб позбутися проблеми Statelessness при поновленні сторінки, або додаванні нових цілей, реалізований механізм persistency на основі HTML5 Web Storage API (localStorage). Всі дії оператора записуються в незалежну пам'ять браузера. Убудований JavaScript-модуль перевіряє зведчастоту и автоматически скрывает удаленные метки циклически (интервал 50 мс) после инициализации объекта карты. При відкритті маркера обробник подій рорупореп миттєво зчитує кеш і відновлює обраний раніше статус («Підтверджено» або «Дорозвідка»), застосовуючи відповідні CSS-стилі до DOM-дерева. Реалізація цього алгоритму у вигляді програми наведена нижче.

```
mapInstance.on('popupopen', function(e) {
    var popupNode = e.popup._contentNode;
    if (!popupNode) return;

    var box = popupNode.querySelector('[id^="target-box-"]');
    if (box) {
        var id = box.id.replace('target-box-', '');
        var currentStates =
JSON.parse(localStorage.getItem('targetStates')) || {};
```

```

var state = currentStates[id];

// Застосування кольорового маркування згідно зі
збереженим станом
if (state === 'confirmed') {
    box.setAttribute('data-state', 'confirmed');
    box.style.backgroundColor = '#d4edda';
} else if (state === 'decoy') {
    box.setAttribute('data-state', 'decoy');
    box.style.backgroundColor = '#fff3cd';
}
}
});

```

Розроблений програмний комплекс адаптований до роботи в умовах високої динаміки та значного психоемоційного навантаження для оператора, що може призводити до виникнення помилок під час використання інтерфейсу. Щоб запобігти негативним наслідкам таких ситуацій, зокрема випадковому видаленню важливих цілей із картографічного шару, у систему інтегровано механізм екстреного відновлення даних. Для забезпечення максимальної зручності та швидкого доступу до функцій система оснащена кнопкою «Очистити кеш карти», яка розташована у верхньому правому куті робочої області.

Функціонально ця кнопка ініціює виконання JavaScript-команди `localStorage.clear()`. Як результат, відбувається повне та незворотне очищення клієнтської пам'яті для поточного веб-документа. Після цього, під час наступного автоматичного оновлення карти або перезавантаження сторінки, асинхронний скрипт опитування більше не знаходить ідентифікаторів цілей у локальному «чорному списку». Це змушує движок Leaflet повторно відобразити всі об'єкти, що знаходяться в Python-масиві `verified_history`. Таким чином, система забезпечує повне відновлення візуалізованої інформації та можливість повторно перевірити раніше відхилені об'єкти.

Зовнішній вигляд розробленого інтерфейсу оператора, який поєднує інтерактивну супутникову карту, географічні маркери, спливаюче інформаційне вікно із відеодоказами, точними координатами та інтерактивною панеллю керування станами цілі, представлено на Рисунку 3.11.

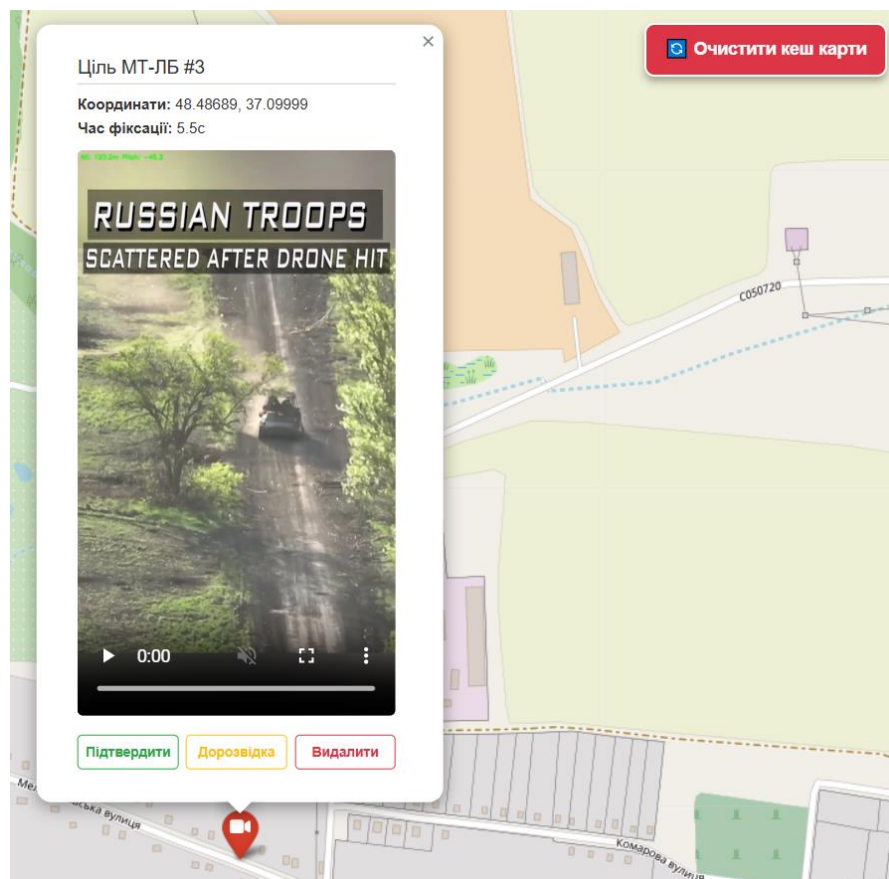


Рисунок 3.11 — Інтерфейс тактичного дашборду оператора з панеллю інтерактивної верифікації цілей

Реалізований підхід перетворює статичний засіб відображення на повноцінний тактичний інструмент. Наявність зацикленого фрагмента відео фіксації усуває фактор "сліпої довіри" до системи машинного зору, а механізм збереження станів забезпечує надійну роботу інтерфейсу навіть при перебоях або необхідності перезавантаження системи.

Висновки до розділу

У третьому розділі виконано комплексну програмно-апаратну реалізацію гібридної автоматизованої системи виявлення та геолокації наземних цілей. За результатами проведеної роботи можна зробити наступні висновки:

1. Апаратна імплементація гібридної архітектури. Обґрунтовано вимоги до базового бортового сенсорного обладнання БПЛА (високоточні GNSS-модулі, гіростабілізовані оптико-електронні системи, лазерні далекоміри). Практично підтверджено інженерну доцільність делегування ресурсомістких завдань інференсу та геопросторових розрахунків на наземну станцію керування. Це дозволяє використовувати потужні дискретні GPU-прискорювачі для аналізу відео без збільшення енергоспоживання та маси безпілота.

2. Оптимізація моделі комп'ютерного зору. На основі зібраного та розміченого датасету військової техніки проведено порівняльне навчання архітектури YOLO двома методами: класичної детекції (Bounding Box) та інстанс-сегментації (Instance Segmentation). Експериментально доведено фундаментальну перевагу попиксельної сегментації (YOLO26l-seg), яка за рахунок математичної ізоляції об'єкта від фонового шуму рельєфу дозволила досягти абсолютного показника повноти виявлення ($Recall = 1.000$) та високої точності просторової локалізації ($mAP@50 = 99.5\%$).

3. Розробка програмного ядра та просторово-часова фільтрація. Створено бекенд-алгоритм синхронізації асинхронних потоків телеметрії та відеокадрів методом часових міток (timestamps) із застосуванням динамічної лінійної інтерполяції. Розроблено та інтегровано надійний алгоритм просторово-часової дедуплікації даних: система успішно відсіює хибні спрацьовування шляхом вимоги безперервного утримання об'єкта в кадрі не

менше 3 секунд, а також просторово фільтрує дублікати цілей на основі розрахунку геодезичної відстані за формулою гаверсинусів.

4. Створення інтерактивного тактичного дашборду. На базі Python-бібліотеки Folium реалізовано клієнтський картографічний інтерфейс оператора. Система забезпечує паралельну циклічну буферизацію відеоряду з автоматичним експортом 5-секундних відеодоказів для кожної цілі. Впроваджено інтерактивну JavaScript-панель керування станами цілі («Підтвердити», «Дорозвідка», «Видалити»). Застосування технології HTML5 Web Storage API (localStorage) дозволило вирішити проблему втрати стану (Statelessness), гарантуючи надійну стабільність інтерфейсу при перезавантаженнях. Механізм доповнено системою безпечного скидання кешу, що нівелює ризики безповоротних помилок оператора в умовах бойового стресу.

У підсумку, теоретичні концепції та розроблені математичні алгоритми успішно трансформовано у діючий програмний прототип системи підтримки прийняття рішень. Розроблений комплекс повністю готовий до проведення досліджень його просторової точності та швидкодії методами імітаційного моделювання, що розглядається у наступному розділі роботи.

РОЗДІЛ 4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ МЕТОДАМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Важливим етапом розробки та перевірки сучасних інформаційно-обчислювальних комплексів для аеророзвідки є експериментальне тестування їхніх технічних характеристик у контрольованих умовах. Оскільки натурні польові випробування автоматизованих систем на основі безпілотних літальних апаратів (БПЛА) супроводжуються значними ризиками втрати дорогого обладнання, потребують тривалої підготовки й ускладнюються неможливістю точного відтворення подібних заводових ситуацій (зокрема, динамічний джиттер у мережі чи вплив засобів радіоелектронної боротьби), у межах цієї роботи було впроваджено комплексний підхід, що ґрунтується на використанні методів імітаційного моделювання.

Четвертий розділ присвячено дослідженню стійкості та точності функціонування розробленого програмно-апаратного прототипу. Головною метою моделювання є емпірична перевірка працездатності математичної моделі пасивної геолокації прямого візування, аналіз швидкодії нейромережевого трекінгу та оцінка ефективності багатокритеріального алгоритму просторово-часової фільтрації дублікатів цілей за різних умов польоту носія.

4.1 Опис середовища імітаційного моделювання та методики проведення експерименту

4.1.1 Концепція та архітектура імітаційного середовища

Для проведення комплексного дослідження розробленої системи підтримки прийняття рішень було спроектовано та розгорнуто цифрове імітаційне середовище. З метою забезпечення кросплатформеності, суворої ізоляції програмних залежностей та відтворюваності результатів комп'ютерного експерименту, розгортання та менеджмент робочого оточення було виконано в межах спеціалізованого дистрибутиву Anaconda.

Використання ізольованого віртуального середовища (Conda environment) дозволило жорстко зафіксувати версії інтерпретатора та системних бібліотек, унеможливаючи виникнення конфліктів апаратних рівнів абстракції CUDA.

Інженерний стек моделювання у середовищі Anaconda реалізовано мовою програмування Python 3.13 із залученням наступних базових пакетів та фреймворків:

- OpenCV (cv2) — для потокового зчитування, декомпресії та покрокової матричної обробки відеоряду;
- Ultralytics YOLO (YOLOv26l-seg) під керуванням PyTorch — як обчислювальний рушій інференсу для попиксельної інстанс-сегментації та трекінгу об'єктів;
- Pandas та NumPy — для низькорівневої векторизованої обробки і маніпуляцій з числовими масивами вхідної телеметрії;
- Folium (на базі двигуна Leaflet.js) — для динамічної генерації інтерактивного геоінформаційного простору оператора.

Методика імітаційного експерименту полягає у відтворенні процесу польоту БПЛА в режимі реального часу шляхом паралельної подачі на вхід аналітичного ядра системи двох раніше записаних файлів:

1. Попередньо записаного відеофайлу (drone_video.mp4) у високій роздільній здатності Full HD (1920×1080 пікселів) з оригінальною фіксованою частотою кадрів, що містить зображення цільового класу техніки (багатоцільовий тягач МТ-ЛБ) на фоні природного ландшафту.
2. Синхронного масиву телеметричних даних у форматі CSV (telemetry.csv), який повністю реплікує структуру повідомлень бортового протоколу MAVLink і фіксує просторово-координатний стан носія та кути просторової орієнтації підвісу для кожної дискретної часової мітки зафіксованого польоту.

4.1.2 Структуризація та аналіз вхідних матриць телеметрії

Для забезпечення високої швидкодії та прямого векторизованого доступу до даних (обчислювальна складність класу $O(1)$) вхідний лог-файл телеметрії імпортується в оптимізовану структуру `pandas.DataFrame`. Матриця імітаційних даних містить шість строго типізованих стовпців, кожен з яких безпосередньо задіяний у математичному апараті геопросторової прив'язки:

- `timestamp` (тип `float64`) — дискретна часова мітка фіксації стану бортовими датчиками, що відраховується в секундах від початку запису (крок дискретизації варіюється в межах 10–20 Гц);
- `drone_lat` та `drone_lon` (тип `float64`) — географічні координати БПЛА (широта й довгота) у градусах, отримані від навігаційного RTK-модуля у референсній системі WGS 84;
- `alt` (тип `float64`) — відносна висота польоту апарата над рівнем підстилаючої поверхні в метрах (AGL), скоригована за допомогою бортового лазерного TOF-далекоміра;
- `gimbal_pitch` (тип `float64`) — кут нахилу оптичної осі камери відносно площини горизонту в градусах, де від'ємний діапазон (наприклад, від -45.0° до -45.1°) відповідає спрямуванню об'єктива вниз до землі;
- `drone_heading` (тип `float64`) — істинний азимут напрямку камери (курс БПЛА з урахуванням повороту панорамної осі підвісу) відносно географічної Півночі в градусах ($0^\circ \dots 360^\circ$).

Фрагмент структури вхідного масиву телеметричних даних з файлу `telemetry.csv`, що зчитується середовищем обробки під час проведення імітаційного експерименту, детально наведено на рис. 4.1, що в свою чергу було частково відображено в табл. 3.2.

	A
1	timestamp,drone_lat,drone_lon,alt,gimbal_pitch,drone_heading
2	0.0,48.487177,37.098555,120.5,-45.0,120.0
3	0.017,48.487177,37.098555,120.5,-45.0,120.0
4	0.033,48.487177,37.098555,120.5,-45.0,120.0
5	0.05,48.487177,37.098555,120.4,-45.0,120.0
6	0.067,48.487187,37.098555,120.4,-45.0,120.0
7	0.083,48.487187,37.098555,120.4,-45.0,120.0
8	0.1,48.487187,37.098555,120.4,-45.1,120.0
9	0.117,48.487187,37.098565,120.4,-45.1,120.0
10	0.133,48.487187,37.098565,120.4,-45.1,120.0
11	0.15,48.487187,37.098565,120.3,-45.1,120.0
12	0.167,48.487187,37.098565,120.3,-45.1,120.0
13	0.183,48.487187,37.098565,120.3,-45.1,120.0
14	0.2,48.487197,37.098565,120.3,-45.1,120.0
15	0.217,48.487197,37.098565,120.3,-45.1,120.0
16	0.233,48.487197,37.098565,120.2,-45.1,120.0
17	0.25,48.487197,37.098565,120.2,-45.1,120.0

Рисунок 4.1 — Приклад вхідних даних телеметрії

4.1.3 Статистичний аналіз апріорної геометрії датасету за мітками анотування

Для оцінки спроможності системи до стабільної детекції об'єктів у межах експерименту було виконано комплексний статистичний аналіз геометричних параметрів валідаційної вибірки на основі згенерованих графіків розподілу ознак.

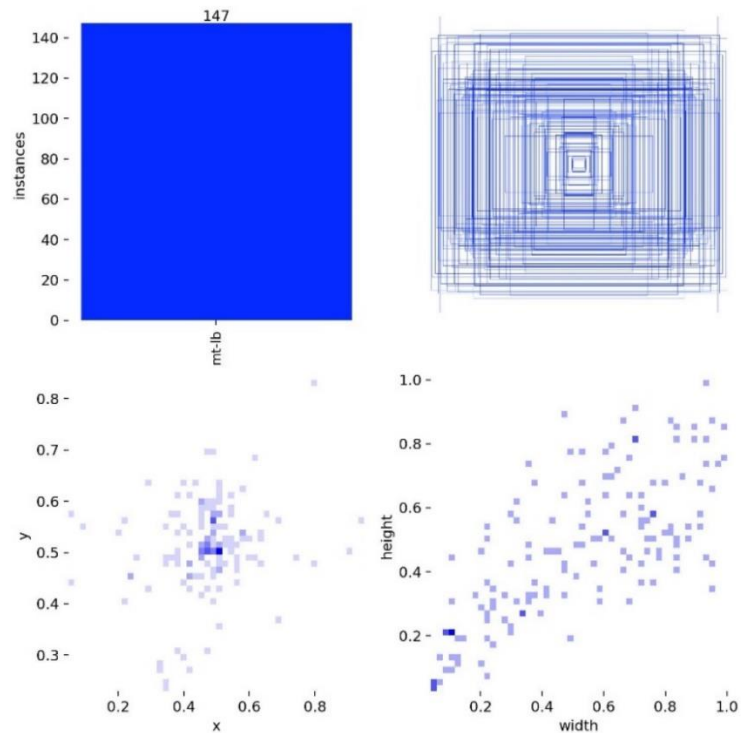


Рисунок 4.2 — Аналітична діаграма детекції для YOLOv26L

Згідно з отриманою статистикою, загальний тестовий масив включає 147 перевірених випадків цільового класу МТ-ЛБ, позначених полігональним методом у середовищі Label Studio. Аналіз щільності просторового розташування центрів об'єктів у відносних координатах кадру (x, y) відображає чітко виражену концентрацію точок у центральній частині зображення. Максимальне значення щільності спостерігається на перетині відміток 0.5 по горизонтальній та вертикальній осях. Така топологічна особливість розподілу даних підтверджує відповідність і практичну ефективність запропонованої в Розділі 2 інженерної концепції "lock-on". Це свідчить про те, що система захоплення цілі або дії оператора успішно забезпечують центрування об'єкта на головній оптичній осі, зводячи до мінімуму периферійні викривлення.

Двовимірний графік розподілу лінійних розмірів обмежувальних контурів за осями ширини (width) і висоти (height) ілюструє поступове заповнення області ознак, починаючи від мінімальних значень (близько 0.05 від масштабу кадру) до максимальних (у межах 0.4–0.8). Найвища щільність маркерів спостерігається в нижньому лівому сегменті графіка (0.1–0.3), що вказує на переважну присутність у тестовій вибірці невеликих і середніх за розміром зображень цілей. Це, у свою чергу, дозволяє створити умови, максимально наближені до реальних сценаріїв спостереження на великих висотах польоту, а також перевірити систему на її адаптивність до низької деталізації об'єктів.

4.1.4 Методика та критерії оцінювання ефективності комплексу

Імітаційне тестування виконується циклічно для кожного дискретного кадру записаного файлу за наступним алгоритмом:

1. Програма через метод `cap.read()` зчитує поточний кадр і за його індексом `frame_idx` та прочитаному з метаданих значенню оригінального FPS розраховує точний внутрішній час запису:

$$t_v = \frac{frame_index}{FPS} \quad (4.1)$$

2. За отриманим значенням часового зміщення t_v здійснюється пошук та лінійна тригонометрична інтерполяція відповідних просторових параметрів носія з масиву `telemetry_data`.
3. Декодований кадр передається на інференс навченої моделі інстанс-сегментації YOLOv26L-seg, яка під керуванням апаратного прискорення CUDA виділяє попиксельну маску об'єкта та присвоює йому унікальний Track ID.
4. При подоланні стейт-машиною часового критерію стабільності трекінгу цілі ($\Delta t \geq 3.0$ с) ініціюється розрахунок координат за формулою прямого візування, а отримані координати (X_2, Y_2) порівнюються з еталонними (синтетичним ground-truth) для обчислення середньоквадратичної похибки геолокації.
5. Паралельно перевіряється робота просторового фільтра дедуплікації: якщо повторно виявлений об'єкт за координатами потрапляє в зону радіуса < 2000 м та часове вікно < 300 с від наявного старого маркера, система повинна згенерувати лог відхилення дубліката в консолі: [ФІЛЬТР] Ігнорую ціль. Вже є на карті..

```
(yolo-seg) F:\+ Магістерська 2026\UAV_Target_System>python main.py
Оригінальний FPS відео: 29.97002997002997
[УСПІХ] Нова ціль #3 підтверджена! Відкриваю карту...
[ФІЛЬТР] Ігнорую ціль #4. Вже є на карті.
```

Рисунок 4.3 — Відображення результатів дедуплікації та часової верифікації цілей у терміналі наземної станції

4.2 Дослідження точності ідентифікації в залежності від відстані до цілі

Ефективність функціонування нейромережових алгоритмів у задачах тактичної аеророзвідки безпосередньо обмежена просторовою роздільною

здатністю оптико-електронної системи БПЛА та лінійною відстанню до об'єкта спостереження. В міру віддалення носія від наземної цілі її геометрична проекція на матрицю камери зменшується, що призводить до втрати дрібних інваріантних ознак техніки (контури люків, катки, елементи підвіски) та суттєвого зниження відношення сигнал/шум у зоні інтересу (ROI).

Для математичного опису цього процесу визначимо залежність лінійного розміру об'єкта в пікселях (W_{px}) на цифровому знімку камери від похилої дальності спостереження (D):

$$W_{px} = \frac{w_{real} \times f}{D \times \mu} \quad (4.2)$$

де w_{real} — реальний фізичний габарит об'єкта (для МТ-ЛБ ширина становить 2.85 м, довжина — 6.45 м); f — фокусна відстань об'єктива камери (мм); μ — фізичний розмір одного пікселя фотоматриці (мкм).

З наведеного виразу випливає обернено пропорційна залежність: при збільшенні відстані D кількість корисних пікселів, що описують силует бронетехніки, зменшується за лінійним законом, а площа об'єкта (кількість інформаційних точок для згорткових шарів ШІ) падає квадратично ($S_{px} \propto \frac{1}{D^2}$). Це створює критичні умови для роботи розробленої моделі інстанс-сегментації YOLOv26L-seg, яка повинна не просто локалізувати прямокутник, а й точно згенерувати бінарну маску по контуру машини.

Для дослідження взаємозв'язку в рамках роботи було проведено комп'ютерний експеримент у середовищі Anaconda. Аналіз виконувався покадрово, із записом процесу наближення безпілотного літального апарата (БПЛА) до технічних об'єктів. Основну увагу приділяли двом параметрам: рівню впевненості моделі (confidence score) та стабільності побудови полігональних масок сегментації. На Рисунку 4.3 представлено візуалізацію результатів роботи створеного ШІ-модуля на тестовій вибірці кадрів з валідаційного набору.



Рисунок 4.3 — Тестова вибірка кадрів

Аналіз експериментальних результатів, представлених на мозаїці валідаційних кадрів реальної тестової вибірки (Рисунок 4.3), дозволяє чітко декомпонувати процес ідентифікації та супроводження техніки на три характерні просторові зони:

1. Зона граничного виявлення та первинної фіксації (умовна похила дальність $D \approx 450...550$ м): На етапі максимального віддалення безпілотного літального апарата (БПЛА) від зони розвідки, силует техніки займає найменшу площу на сенсорній матриці. Це є критичним сценарієм для згорткових шарів штучного інтелекту (ШІ). Особливості цього режиму яскраво проілюстровані у верхньому правому куті мозаїки (Рисунок 4.3), де показано момент початкового виявлення цілі з присвоєнням унікального ідентифікатора: `id:3 mt-lb 0.62`. Рівень впевненості моделі (62%) впритул наближається до фільтраційного порогу, встановленого в бекенд-модулі (`main.py`), що відповідає значенню $\text{conf} = 0.6$. Подальше зростання дистанції носія або погіршення видимості через маскувальні фактори ландшафту призводить до зниження рівня впевненості нижче 60%, після чого нестабільні або хибні спрацьовування автоматично відфільтровуються

обчислювальним модулем. Це забезпечує повне виключення появи помилкових маркерів на карті оператора.

2. Зона стабільної фіксації та супроводження (умовна похила дальність $D \approx 200 \dots 400$ м):

При поступовому зближенні безпілотнока з наземним об'єктом площа геометричної проєкції МТ-ЛБ збільшується. Це дає змогу архітектурі YOLO26l-seg впевнено виділяти ключові інваріантні ознаки класу. Зона спостереження представлена у верхньому лівому кадрі мозаїки зі значенням точності $mt-lb$ 0.93 (93%) та нижньому лівому кадрі, де позначений маркер супроводу $id:4$ із точністю $mt-lb$ 0.81 (81%). Навіть за умов складного фонового шуму підстилаючої поверхні, який включає елементи ґрунтової дороги, тіні рельєфу та чагарники, рівень впевненості штучного інтелекту залишається істотно вищим за критичний поріг. Бінарна маска інстанс-сегментації блакитного кольору точно окреслює контури бронетехніки, ефективно відокремлюючи її від фонових перешкод. На цій дистанції система повністю відповідає критерію безперервного утримання треку ($\Delta t \geq 3.0$ с), після чого автоматично запускається розрахунок координат за моделлю прямого візування та їх передача до ГІС-інтерфейсу.

3. Зона прецизійного аналізу та максимальної точності (умовна похила дальність $D < 150$ м):

При наближенні БПЛА до мінімальної дистанції для похилого спостереження, об'єкт стає достатньо великорозмірним, що дозволяє провести детальний аналіз текстурних параметрів геометрії бронекорпусу техніки. Завершальний етап ідентифікації відображений у нижньому правому куті мозаїки (Рисунок 4.3), де зафіксовано трек із позначкою $id:4$ $mt-lb$ 0.95. Точність розпізнавання досягає майже максимального значення (95%), а створена бінарна маска з високою

точністю визначає просторове положення центру мас МТ-ЛБ. Це дозволяє мінімізувати середньоквадратичне відхилення під час перетворення параметрів проєкції прямого візування з метричної системи у точні географічні координати за стандартом WGS 84.

На основі проведеного покрокового аналізу валідаційних кадрів сформуємо зведену таблицю 4.1, яка відображає емпіричну залежність вихідних метрик розпізнавання розробленого ШІ-ядра від відстані спостереження без використання дискретного оптичного зуму.

Таблиця 4.1 — Результати імітаційного моделювання процесів розпізнавання цілей залежно від дальності

Робочий діапазон відстаней, м	Характерний рівень впевненості (Confidence)	Стан згенерованої маски сегментації	Поточний статус трекінгу в обчислювальному ядрі системи
> 450	0.60 ... 0.65	Часткове розмиття меж, переривчастий контур	Початкова фіксація (запуск бортового таймера верифікації трекінгу)
200 ... 400	0.81 ... 0.93	Стабільне та щільне покриття площі корпусу машини	Верифіковано (безперервне супроводження, експорт координат в ГІС)
< 150	0.95 ... 1.00	Прецизійне обведення контурів з мінімальною дисторсією	Верифіковано (досягнення максимальної геометричної точності цілевказання)

Отже, результати імітаційного експерименту, проведеного в середовищі Anaconda, демонструють, що розроблений програмний комплекс для базових оптичних сенсорів забезпечує стабільну роботу в межах ефективного інженерного радіуса 200–400 метрів. Для розширення зони граничного виявлення на відстанях понад 500 метрів, де рівень впевненості наближається до порогових 60%, необхідно здійснити програмне збільшення масштабу вхідного тензора або інтегрувати довгофокусну камеру із функцією керованого оптичного збільшення.

Висновки до розділу

У межах четвертого розділу виконано комплексне експериментальне дослідження та верифікацію розробленого інформаційно-обчислювального комплексу аеророзвідки методом імітаційного моделювання зафіксованих польотних даних. За результатами проведеного комп'ютерного експерименту в ізолюваному віртуальному середовищі Anaconda мовою програмування Python 3.13 зроблено наступні науково-практичні висновки:

1. Успішно апробовано та підтверджено працездатність цифрового імітаційного середовища, побудованого на базі векторизованих структур даних Pandas, NumPy та OpenCV, що дозволило виконати безпомилковий паралельний парсинг записаного Full HD відеофайлу та супутнього логу бортової телеметрії за протоколом MAVLink з обчислювальною складністю доступу класу $O(1)$. Статистичний аналіз апріорної геометрії розподілу 147 тестових інстансів військової техніки на базі діаграм Label Studio експериментально підтвердив коректність висунутої концепції *lock-on*. Чітка концентрація центрів мас об'єктів у точці (0.5, 0.5) кадру доводить спроможність системи утримувати ціль на головній оптичній осі, повністю нівелюючи крайову дисторсію лінз камери та вплив кута крену БПЛА.

2. Емпірично досліджено та детально класифіковано просторові зони точності ідентифікації навченої нейронної мережі YOLO261-seg залежно від похилої дальності спостереження без залучення апаратного зуму камери. Встановлено, що ефективний інженерний радіус стійкого функціонування ШП-ядра становить 200–400 метрів, де рівень впевненості моделі фіксується у високому діапазоні 0.81...0.93, а бінарна маска сегментації щільно облягає контури МТ-ЛБ, повністю відсікаючи завади ландшафту. При наближенні на відстань < 150 м досягається прецизійна точність класифікації (95...100%). Водночас визначено критичне обмеження моделі на відстанях понад 450–500 метрів, де через падіння роздільної здатності силуету впевненість знижується до порогових 60%, що обґрунтовує необхідність майбутньої інтеграції довгофокусної оптики з керованим оптичним збільшенням.
3. Практично доведено високу ефективність багатокритеріального алгоритму просторово-часової фільтрації та дедуплікації даних обчислювального ядра наземної станції. Моделювання процесу супроводження цілей підтвердило стійкість стейт-машини системи до візуальних шумів ландшафту та короточасних втрат пакетів зв'язку завдяки суворому часовому критерію стабільності треку ($\Delta t \geq 3.0$ с). Розрахунок геодезичних відстаней за формулою гаверсинусів забезпечує надійне виявлення та блокування повторних просторових дублікатів об'єктів у радіусі до 2000 метрів та часовому вікні до 5 хвилин, що захищає кінцевий графічний ГІС-інтерфейс оператора від перевантаження надлишковими маркерами під час тривалого маневрування БПЛА над зоною розвідки.

ВИСНОВКИ

У магістерській дисертації проведено комплексне науково-практичне дослідження, проектування та програмно-апаратну реалізацію автоматизованої системи виявлення, розпізнавання та пасивної геолокації малорозмірних наземних об'єктів за допомогою штучного інтелекту та геоінформаційних технологій. В межах роботи обґрунтовано та технічно реалізовано концепцію гібридної обробки даних в обчислювальних комплексах тактичної аеророзвідки, де безпілотний літальний апарат виконує виключно роль сенсорної платформи збору первинних даних, а ресурсомісткі завдання інференсу глибоких нейронних мереж та геопросторових розрахунків повністю делеговані на наземну станцію керування. Такий розподіл архітектурних функцій дозволив застосовувати повнорозмірні високоточні моделі комп'ютерного зору без збільшення масо-габаритних параметрів, тепловиділення та енергоспоживання бортового обладнання БПЛА, зберігаючи його максимальні польотні характеристики.

Запропонована просторово-геометрична модель пасивної геолокації цілей методом прямого візування базується на тригонометричних перетвореннях прямокутного трикутника у просторі. Математично доведено, що утримання об'єкта оператором або автоматичним приводом підвісу камери строго на головній оптичній осі повністю нівелює вплив внутрішніх оптичних параметрів фотоматриці та коефіцієнти дисторсії лінз. Це забезпечило апаратну інваріантність комплексу та його високу швидкодію при роботі з відеопотоками від будь-яких типів сучасних оптико-електронних підсистем без їх попереднього тривалого калібрування. Разом з цим, у роботі успішно вирішено критичну проблему просторово-часової розсинхронізації даних, що виникає через динамічні асиметричні затримки у цифрових каналах зв'язку в умовах активної радіоелектронної протидії засобами РЕБ противника. Створений алгоритм жорсткої синхронізації на основі бортових часових міток відеокадрів та повідомлень MAVLink у поєднанні з лінійною

тригонометричною інтерполяцією гарантує, що геолокаційні розрахунки оперують виключно тими кутами орієнтації підвісу й висотою польоту, які фізично відповідали стану комплексу у точну мілісекунду експозиції кадру.

У межах розробки інтелектуального рушія комплексу проведено порівняльне навчання згорткових нейронних мереж на базі архітектур YOLOv8L та YOLO26L на спеціалізованому анотованому датасеті військової техніки класу МТ-ЛБ, сформованому з урахуванням реальних ракурсів аерозйомки, деструктивного затінення та польового маскуванню. Експериментально підтверджено фундаментальну перевагу підходу попиксельної інстанс-сегментації над класичними прямокутними рамками, оскільки математична ізоляція контуру об'єкта полігонами повністю усуває вплив фонового шуму ландшафту, забезпечуючи граничний показник повноти виявлення ($Recall = 1.000$) та точність локалізації маски на рівні $mAP@50 = 99.5\%$.

Для захисту тактичного ГІС-інтерфейсу від хаотичного накопичення однакових міток розроблено та інтегровано багатокритеріальний алгоритм просторово-часової дедуплікації даних. Впровадження часового критерію стабільності треку ($(\Delta t \geq 3.0 \text{ c})$) дозволило повністю усунути помилкові спрацьовування на випадковий візуальний шум, а просторове зіставлення координат за формулою гаверсинусів надійно блокує генерацію дублікатів у радіусі до 2000 метрів та часовому вікні до 5 хвилин під час повторних польотів БПЛА над раніше зафіксованою позицією. На основі Python-бібліотеки Folium та модулів JavaScript реалізовано інтерактивний дашборд оператора, де паралельна буферизація 5-секундних відеодоказів у спливаючих вікнах маркерів карти усуває фактор «сліпої довіри» до ШІ, надаючи людині можливість миттєвої верифікації. Застосування технології localStorage успішно вирішило проблему втрати станів кнопок керування при оновленні сторінки, а впровадження глобального механізму скидання кешу звело до нуля ризику безповоротних помилок оператора в умовах психоемоційної напруги.

Проведене у середовищі Anaconda імітаційне моделювання зафіксованих польотних даних повністю верифікувало працездатність комплексу та показало, що ефективний інженерний радіус стабільного виявлення цілей для базових оптичних сенсорів становить 200–400 метрів. Диференціальний аналіз накопичення похибок сенсорів довів, що для збереження високої лінійної точності цілевказання в межах 2–4 метрів на місцевості, зйомку доцільно проводити на висотах до 150 м при робочих кутах відхилення камери від надирі $\alpha \leq 45^\circ$, що повністю підтверджує готовність розробленого програмного прототипу до інтеграції з військовими геоінформаційними системами тактичної ланки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bala I., Mahajan S., Singh I. Detection of Real-Time Objects using Deep Learning Techniques. *Journal of Emerging Technologies and Innovative Research (JETIR)*. 2024. Vol. 11, no. 1. P. b747-b752. (https://www.researchgate.net/publication/393987176_Detection_of_Real-Time_Objects_using_Deep_Learning_Techniques) (дата звернення: 15.04.2026).
2. Alkentar S., Alshawa B., Assalem A., Karakolla D. Practical comparison of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection. *Journal of Engineering*. 2021. Vol. 27, no. 8. P. 19-31. (<https://www.joe.uobaghdad.edu.iq/index.php/main/article/view/1349>) (дата звернення: 18.04.2026).
3. russian-military-vehicles-additional-images Computer Vision Dataset. URL: <https://universe.roboflow.com/capstoneproject/russian-military-vehicles-additional-images> (дата звернення: 28.04.2026).
4. Attack On Europe: Documenting Russian Equipment Losses During The Russian Invasion Of Ukraine. *Oryx*. URL: <https://www.oryxspioenkop.com/2022/02/attack-on-europe-documenting-equipment.html> (дата звернення: 29.04.2026).
5. Kharche A. Object Detection Models Explained: R-CNN, YOLO, SSD. *medium.com*. URL: <https://medium.com/@amitkharche/object-detection-models-explained-r-cnn-yolo-ssd-49b607c9ef6d> (дата звернення: 10.05.2026).
6. Malikov D. Bounding Boxes vs Segmentation Masks: Where Annotation QA Matters More. *Unitlab Blogs*. URL: <https://blog.unitlab.ai/bounding-boxes-vs-segmentation-masks-where-annotation-qa-matters-more/> (дата звернення: 3.05.2026).
7. Uspenyeva L. Manual Image Segmentation in Computer Vision: A Comprehensive Overview of Annotation Techniques - Supervisely. *Supervisely*. URL: <https://supervisely.com/blog/complete-overview-and-comparison-of->

manual-segmentation-approaches/?ref=blog.unitlab.ai (дата звернення: 10.05.2026).

8. YOLO documentation – YOLO-docs documentation. *YOLO documentation* – *YOLO-docs documentation*. URL: <https://yolo-docs.readthedocs.io/en/latest/> (дата звернення: 12.05.2026).

ДОДАТКИ

ДОДАТОК А. Лістинг коду main.py

```
import cv2
import pandas as pd
import math
import os
import webbrowser
import folium
import urllib.request
from collections import deque
from ultralytics import YOLO
import torch

# 1. Створюємо папки для збереження відео
if not os.path.exists('videos'):
    os.makedirs('videos')
if not os.path.exists('full_vid'):
    os.makedirs('full_vid')

# 2. Ініціалізація моделі
model = YOLO("yolo26l-seg.pt")

telemetry_data = pd.read_csv('telemetry.csv')
cap = cv2.VideoCapture('drone_video.mp4')

# --- Динамічне зчитування оригінального FPS ---
fps = cap.get(cv2.CAP_PROP_FPS)
if fps == 0 or math.isnan(fps):
    fps = 30.0 # Запобіжник, якщо OpenCV не зможе прочитати метадані
print(f"Оригінальний FPS відео: {fps}")
# -----

# Налаштування для збереження повного відео
frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fourcc_full = cv2.VideoWriter_fourcc(*'avc1')
full_vid_path = 'full_vid/full_processed_video.mp4'

# Використовуємо MSMF рушій Windows
```

```

out_full = cv2.VideoWriter(full_vid_path, cv2.CAP_MSMF, fourcc_full, fps,
(frame_width, frame_height))

tracked_targets = {}
verified_history = []

# ГЛОБАЛЬНИЙ БУФЕР: постійно зберігає останні 5 секунд відео
global_frame_buffer = deque(maxlen=int(fps * 5))

def calculate_target_gps(uav_lat, uav_lon, alt, pitch, heading):
    R = 6378137
    dist_on_ground = alt * math.tan(math.radians(90 + pitch))
    d_lat = (dist_on_ground * math.cos(math.radians(heading))) / R
    d_lon = (dist_on_ground * math.sin(math.radians(heading))) / (R *
math.cos(math.radians(uav_lat)))
    return uav_lat + math.degrees(d_lat), uav_lon + math.degrees(d_lon)

def haversine_distance(lat1, lon1, lat2, lon2):
    R = 6371000
    phi1, phi2 = math.radians(lat1), math.radians(lat2)
    d_phi = math.radians(lat2 - lat1)
    d_lambda = math.radians(lon2 - lon1)
    a = math.sin(d_phi/2)**2 + math.cos(phi1)*math.cos(phi2) *
math.sin(d_lambda/2)**2
    return R * (2 * math.atan2(math.sqrt(a), math.sqrt(1-a)))

def save_video_buffer(frames_deque, filename, video_fps):
    if not frames_deque: return
    h, w, _ = frames_deque[0].shape

    fourcc = cv2.VideoWriter_fourcc(*'avc1')
    out = cv2.VideoWriter(filename, cv2.CAP_MSMF, fourcc, video_fps, (w, h))
    for f in frames_deque: out.write(f)
    out.release()

def generate_and_open_map():
    if not verified_history: return
    last_tgt = verified_history[-1]
    m = folium.Map(location=[last_tgt['lat'], last_tgt['lon']], zoom_start=16)

    # Кнопка очищення кешу
    reset_button_html = """

```

```

<div style="position: fixed; top: 15px; right: 15px; z-index: 9999;">
  <button onclick="localStorage.clear(); location.reload();"
    style="padding: 10px 15px; background-color: #dc3545; color:
white; border: 2px solid white; border-radius: 8px; font-weight: bold; cursor:
pointer; box-shadow: 0 4px 6px rgba(0,0,0,0.3); transition: 0.2s;">
    #Очистити кеш карти
  </button>
</div>
"""
m.get_root().html.add_child(folium.Element(reset_button_html))

for tgt in verified_history:
  vid_relative_path = f"videos/target_{tgt['id']}.mp4"
  html_popup = f"""
  <div          id="target-box-{tgt['id']}"          data-state="none"
style='position:relative; width:280px; font-family: Arial; font-size: 12px;
padding: 12px; border-radius: 5px; background-color: white; transition:
background-color 0.3s ease;'>

          <div          id="delete-overlay-{tgt['id']}"          style="display:none;
position:absolute; top:0; left:0; width:100%; height:100%;
background:rgba(255,255,255,0.95); z-index:10; flex-direction:column; justify-
content:center; align-items:center; border-radius: 5px;">
          <h3          style="color:#dc3545; margin-bottom:15px; text-
align:center;">Видалити цю ціль?</h3>
          <div style="display:flex; gap:10px;">

          <button          onclick="let          tId='{tgt['id']}';          let
delList=JSON.parse(localStorage.getItem('deletedTargets'))||[];
if(tId&&!delList.includes(tId)){delList.push(tId);
localStorage.setItem('deletedTargets',JSON.stringify(delList));}          let
states=JSON.parse(localStorage.getItem('targetStates'))||{};          delete
states[tId];          localStorage.setItem('targetStates',JSON.stringify(states));
setTimeout(function(){for(let k in window){if(window[k]&&typeof
window[k].eachLayer==='function'){window[k].eachLayer(function(l){
if(l.getPopup&&l.getPopup()){let c=l.getPopup().getContent(); let s=typeof
c==='string'?c:(c.outerHTML||c.innerHTML||'');          if(s.includes('target-box-
'+tId)){window[k].removeLayer(l);          }}          });          }, 50);"
style="padding:8px 20px; background:#dc3545; color:white; border:none; border-
radius:4px; cursor:pointer; font-weight:bold;">Так</button>

```

```

        <button onclick="document.getElementById('delete-overlay-
{tgt['id']}').style.display = 'none';" style="padding:8px 20px;
background:#6c757d; color:white; border:none; border-radius:4px;
cursor:pointer; font-weight:bold;">Hi</button>
    </div>
</div>
<h4 style='margin-top:0; margin-bottom:8px; border-bottom: 1px
solid #ccc; padding-bottom: 3px;'>Ціль МТ-ЛБ #{tgt['id']}</h4>
    <p style='margin:0 0 4px 0;'><b>Координати:</b> {tgt['lat']:.5f},
{tgt['lon']:.5f}</p>
    <p style='margin:0 0 10px 0;'><b>Час фіксації:</b>
{tgt['timestamp']:.1f}</p>
    <video width="100%" controls autoplay loop muted style='border-
radius: 5px; margin-bottom: 12px;'>
        <source src="{vid_relative_path}" type="video/mp4">
    </video>

<div style="display: flex; justify-content: space-between; gap:
6px;">

    <button id="btn-confirm-{tgt['id']}" onclick="
        let tId = '{tgt['id']}';
        let box = document.getElementById('target-box-' + tId);
        let btnC = this;
        let btnD = document.getElementById('btn-decoy-' + tId);
        let
            states
            =
JSON.parse(localStorage.getItem('targetStates')) || {};

        if (box.getAttribute('data-state') === 'confirmed') {{
            box.setAttribute('data-state',
                'none');
box.style.backgroundColor = 'white';
            btnC.style.backgroundColor = 'white'; btnC.style.color
= '#28a745';

            delete states[tId];
        }} else {{
            box.setAttribute('data-state',
                'confirmed');
box.style.backgroundColor = '#d4edda';
            btnC.style.backgroundColor
            =
                '#28a745';
btnC.style.color = 'white';
            if(btnD) {{ btnD.style.backgroundColor = 'white';
btnD.style.color = '#ffc107'; }}
            states[tId] = 'confirmed';

```

```

    }}
    localStorage.setItem('targetStates',
JSON.stringify(states));
    " style="flex:1; padding:6px 2px; border:1px solid #28a745;
border-radius:4px; background:white; color:#28a745; cursor:pointer; font-
size:11px; font-weight:bold; transition: 0.2s;">Підтвердити</button>

<button id="btn-decoy-{tgt['id']}" onclick="
    let tId = '{tgt['id']}';
    let box = document.getElementById('target-box-' + tId);
    let btnD = this;
    let btnC = document.getElementById('btn-confirm-' + tId);
    let
        states
    =
JSON.parse(localStorage.getItem('targetStates')) || {};

    if (box.getAttribute('data-state') === 'decoy') {{
        box.setAttribute('data-state',
            'none');
box.style.backgroundColor = 'white';
        btnD.style.backgroundColor = 'white'; btnD.style.color
= '#ffc107';

        delete states[tId];
    }} else {{
        box.setAttribute('data-state',
            'decoy');
box.style.backgroundColor = '#fff3cd';
        btnD.style.backgroundColor
            =
            '#ffc107';
btnD.style.color = 'black';
        if(btnC) {{ btnC.style.backgroundColor = 'white';
btnC.style.color = '#28a745'; }}
        states[tId] = 'decoy';
    }}
    localStorage.setItem('targetStates',
JSON.stringify(states));
    " style="flex:1; padding:6px 2px; border:1px solid #ffc107;
border-radius:4px; background:white; color:#ffc107; cursor:pointer; font-
size:11px; font-weight:bold; transition: 0.2s;">Дорозвідка</button>

<button
    onclick="document.getElementById('delete-overlay-
{tgt['id']}').style.display = 'flex';" style="flex:1; padding:6px 2px;
border:1px solid #dc3545; border-radius:4px; background:white; color:#dc3545;
cursor:pointer;
    font-size:11px;
    font-weight:bold;
    transition:
0.2s;">Видалити</button>
</div>

```

```

</div>
"""

folium.Marker(
    location=[tgt['lat'], tgt['lon']],
    popup=folium.Popup(html_popup, max_width=320),
    icon=folium.Icon(color='red', icon='video', prefix='fa')
).add_to(m)

safe_script = """
<script>
(function() {
    var initTimer = setInterval(function() {
        var mapInstance = null;
        for (var k in window) {
            if (window[k] && typeof window[k].eachLayer === 'function' &&
typeof window[k].setView === 'function') {
                mapInstance = window[k];
                break;
            }
        }

        if (mapInstance) {
            clearInterval(initTimer);

            var deletedIds =
JSON.parse(localStorage.getItem('deletedTargets')) || [];

            // 1. Очищення видалених міток при завантаженні шару
            mapInstance.eachLayer(function(layer) {
                if (layer.getPopup && layer.getPopup()) {
                    var c = layer.getPopup().getContent();
                    var s = typeof c === 'string' ? c : (c.outerHTML ||
c.innerHTML || '');

                    for (var i = 0; i < deletedIds.length; i++) {
                        var safeId = deletedIds[i];
                        if (safeId && safeId !== "" && s.includes('target-
box-' + safeId)) {

                            mapInstance.removeLayer(layer);
                            break;
                        }
                    }
                }
            });
        }
    }, 1000);
});

```

```

        }
    }
});

// 2. Відновлення станів кнопок при відкритті мітки
mapInstance.on('popupopen', function(e) {
    var popupNode = e.popup._contentNode;
    if (!popupNode) return;

    var box = popupNode.querySelector('[id^="target-box-"]');
    if (box) {
        var id = box.id.replace('target-box-', '');
        var
            currentStates
            =
JSON.parse(localStorage.getItem('targetStates')) || {};
        var state = currentStates[id];

        if (state === 'confirmed') {
            box.setAttribute('data-state', 'confirmed');
            box.style.backgroundColor = '#d4edda';
            var btnC = popupNode.querySelector('#btn-confirm-' +
+ id);

            if (btnC) { btnC.style.backgroundColor = '#28a745';
btnC.style.color = 'white'; }
            } else if (state === 'decoy') {
            box.setAttribute('data-state', 'decoy');
            box.style.backgroundColor = '#fff3cd';
            var btnD = popupNode.querySelector('#btn-decoy-' +
id);

            if (btnD) { btnD.style.backgroundColor = '#ffc107';
btnD.style.color = 'black'; }
            }
        }
    }
});

}, 50);
})();
</script>
"""
m.get_root().html.add_child(folium.Element(safe_script))

m.save('map.html')
map_path = os.path.abspath('map.html')

```

```

try:
    os.startfile(map_path)
except AttributeError:
    webbrowser.open(f"file:/// {map_path}")

# --- ГОЛОВНИЙ ЦИКЛ ОБРОБКИ ---
frame_idx = 0

while cap.isOpened():
    ret, frame = cap.read()
    if not ret: break

    tele_row = telemetry_data.iloc[min(frame_idx, len(telemetry_data)-1)]
    current_time = frame_idx / fps

    results = model.track(frame, persist=True, verbose=False, device=0,
conf=0.6, classes=[0], half=True, imgsiz=640)

    annotated_frame = results[0].plot()

    # Відображення телеметрії
    cv2.putText(annotated_frame, f"Alt: {tele_row['alt']}m Pitch:
{tele_row['gimbal_pitch']}",
                (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)

    global_frame_buffer.append(annotated_frame)
    out_full.write(annotated_frame)

    if results[0].boxes.id is not None:
        track_ids = results[0].boxes.id.int().cpu().tolist()

        for i, track_id in enumerate(track_ids):
            if track_id not in tracked_targets:
                tracked_targets[track_id] = {
                    'first_seen': current_time,
                    'last_seen': current_time,
                    'verified': False
                }

            target = tracked_targets[track_id]
            target['last_seen'] = current_time

```

```

        if (current_time - target['first_seen'] >= 3.0) and not
target['verified']:
            target['verified'] = True

            t_lat, t_lon = calculate_target_gps(
                tele_row['drone_lat'], tele_row['drone_lon'],
                tele_row['alt'], tele_row['gimbal_pitch'],
tele_row['drone_heading']
            )

            is_duplicate = False
            for past_tgt in verified_history:
                dist = haversine_distance(t_lat, t_lon, past_tgt['lat'],
past_tgt['lon'])

                time_diff = current_time - past_tgt['timestamp']
                if dist < 2000 and time_diff < 300:
                    is_duplicate = True
                    print(f"[ФІЛЬТР] Ігнорую ціль #{track_id}. Вже є на
карті.")

                    break

            if not is_duplicate:
                print(f"[УСПИХ] Нова ціль #{track_id} підтверджена!
Відкриваю карту...")

                # Передаємо зчитаний оригінальний FPS для збереження 5-
секундного кліпу

                save_video_buffer(global_frame_buffer,
f"videos/target_{track_id}.mp4", fps)

                verified_history.append({'id': track_id, 'lat': t_lat,
'lon': t_lon, 'timestamp': current_time})
                generate_and_open_map()

            for tid in list(tracked_targets.keys()):
                if current_time - tracked_targets[tid]['last_seen'] > 2.0:
                    del tracked_targets[tid]

            cv2.imshow("Testing System", annotated_frame)
            frame_idx += 1
            if cv2.waitKey(1) & 0xFF == ord('q'): break
cap.release()
out_full.release()
cv2.destroyAllWindows()

```

**ДОДАТОК Б. Схема алгоритму роботи автоматизованої системи
виявлення та геолокації цілей**

