

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

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

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Програмний комплекс виявлення та програмного захоплення цілей для
FPV-дронів в умовах обмежених обчислювальних ресурсів
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-2

_____ Іван ТУРЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник доц. Марія ГОЛОВЯНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

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

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Туренку Івану Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмний комплекс виявлення та програмного захоплення цілей для FPV-дронів в умовах обмежених обчислювальних ресурсів _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 липня 2026 р.

3. Вихідні дані до роботи Науково-технічні публікації в галузі комп'ютерного зору та супроводження цілей БпЛА, технічна документація мови Python, бібліотек PyTorch, OpenCV та Ultralytics, специфікації обчислювальних платформ Edge AI (зокрема NVIDIA Jetson), відеоматеріали польотів у симуляторі Remote Reaper, набір графічних даних із зображеннями військової техніки для навчання та тестування моделей.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі дослідження _____

2) Теоретико-методологічні основи оптимізації даних та архітектури моделей _____

3) Програмно-алгоритмічна реалізація конвеєра підготовки даних та системи детекції _____

4) Експериментальні дослідження та абляційний аналіз системи _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	12.06.2026	виконано
3	Огляд існуючих метрик оцінювання якості	15.06.2026	виконано
4	Аналіз підходів вирішення задачі детекції	19.06.2026	виконано
5	Експериментальне моделювання та навчання моделі	25.06.2026	виконано
6	Написання пояснювальної записки	29.06.2026	виконано
7	Перевірка на академічний плагіат	30.06.2026	виконано
8	Нормоконтроль	06.07.2026	виконано
9	Підготовка презентації та доповіді	08.07.2026	виконано
10	Попередній захист	10.07.2026	виконано
11	Рецензування	11.07.2026	виконано
12	Захист перед ЕК	14.07.2026	виконано

Дата видачі завдання 8 червня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

доц. Марія ГОЛОВЯНКО
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 103 с., 11 рис., 7 табл., 10 дод., 50 джерел.

АВТОНОМНЕ НАВЕДЕННЯ, ВІЗУАЛЬНИЙ ТРЕКІНГ, КОМП'ЮТЕРНИЙ ЗІР, ОБМЕЖЕНІ РЕСУРСИ, ОПТИМІЗАЦІЯ МОДЕЛЕЙ, EDGE AI, FPV-ДРОН, ОС-SORT, YOLO26-NANO.

Об'єкт дослідження – процес реального розпізнавання та супроводження об'єктів у відеопотоці.

Предмет дослідження – методи оптимізації та зниження розрядності ваг моделей YOLO і алгоритми трекінгу для Edge-платформ.

Мета роботи – розробка та оптимізація програмного комплексу автономного захоплення та утримання цілей БпЛА в реальному часі.

Методи дослідження – комп'ютерний зір, фільтрація Калмана, квантування ваг моделей, мова Python з бібліотеками OpenCV та PyTorch.

У роботі розроблено програмний комплекс автономного виявлення та супроводження рухомих цілей БпЛА в реальному часі. Оптимізацію обсягу даних проведено за допомогою алгоритму KASFraD, напівавтоматичної розмітки SAM3 та дедуплікації dHash з фільтрацією розміття за Лапласіаном, що дозволило зменшити вибірку на 50.5% (до 5735 кадрів) при балансі класів 90% на 10%. Запропоновано конвеєр на базі детектора YOLO26-nano та трекера ОС-SORT з асинхронною трипотоковою архітектурою. Перехід до напівточності FP16 знизив затримку інференсу до 1.81 мс та підвищив швидкодію до 552.7 FPS при мінімальній втраті mAP50 на 0.24% (до 0.7563). Трекер ОС-SORT продемонстрував стійкість до зміни ракурсів та перекриттів об'єктів з оцінкою НОТА 52.12%. Загальна латентність обробки кадру становить 2.761 мс, що задовольняє вимоги реального часу. Результати рекомендовано для впровадження на вбудованих платформах сімейства NVIDIA Jetson Orin Nano.

ABSTRACT

Bachelor's thesis contains: 103 pp., 11 fig., 7 tabl., 10 ann., 50 references.

AUTONOMOUS GUIDANCE, COMPUTER VISION, EDGE AI, FPV DRONE, LIMITED RESOURCES, MODEL OPTIMIZATION, OC-SORT, VISUAL TRACKING, YOLO26-NANO.

Object of research – the process of real-time automatic detection and tracking of objects in a video stream.

Subject of research – YOLO neural network optimization methods, weight precision reduction, and tracking algorithms for Edge platforms.

Goal of the work – development and optimization of a real-time target detection and tracking software system for autonomous target lock on UAVs.

Methods of research – computer vision, Kalman filtering, model quantization, Python language with OpenCV and PyTorch libraries.

The thesis develops a software system for real-time autonomous detection and tracking of moving targets for UAVs. Data optimization is achieved using the KASFraD algorithm, SAM3 semi-automated labeling, and dHash deduplication with Laplacian blur filtering, reducing the dataset by 50.5% (to 5,735 frames) with a 90% to 10% class balance. A pipeline combining the YOLO26-nano detector and the OC-SORT tracker with an asynchronous three-threaded architecture is proposed. Transitioning to FP16 precision reduced inference latency to 1.81 ms and increased throughput to 552.7 FPS, with a minor mAP50 loss of 0.24% (to 0.7563). The OC-SORT tracker demonstrated robustness to angle changes and occlusions with a HOTA score of 52.12%. The total frame processing latency of 2.761 ms meets real-time requirements. The results are recommended for deployment on NVIDIA Jetson Orin Nano embedded platforms.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	11
Вступ.....	13
1 Аналіз предметної галузі та постановка задачі дослідження	15
1.1 Специфіка предметної галузі (FPV-пілотування).....	15
1.2 Порівняльний аналіз наявних систем автоматичного захоплення та супроводження цілей	16
1.3 Системні обмеження Edge-платформ та вимоги до обробки даних у реальному часі (SWaP-C)	18
1.4 Дослідження методів детекції та алгоритмів супроводження об'єктів в умовах доменного зсуву даних	21
1.4.1 Дослідження методів комп'ютерного зору для детекції (огляд родин моделей, вибір на користь YOLO-подібних).....	21
1.4.2 Огляд наявних алгоритмів трекінгу для утримання цілі	24
1.4.3 Аналіз даних та методів оцінки (особливості FPV-зображень, огляд датасетів, метрики mAP, IoU)	26
1.4.4 Апаратні рішення та методи їх оптимізації (огляд Edge платформ, квантування під слабе апаратне забезпечення).....	29
1.5 Постановка науково-практичної задачі та обґрунтування критеріїв оцінки ефективності розробки	32
1.5.1 Обґрунтування критеріїв оцінки ефективності	33
2 Теоретико-методологічні основи оптимізації даних та архітектури моделей.....	35
2.1 Математична модель кінематично-адаптивного просторово-часового проріджування часових рядів за методом KASFraD	35
2.2 Методологія автоматизованого псевдоанотування (SAM3) та оркестрації хмарних репозиторіїв засобами CVAT REST API	40

2.3 Алгоритмічні засади евристичної дедуплікації зображень на основі різницевого хешування, фільтрації розмиття та методу вибору найвіддаленіших точок (FPS).....	41
2.3.1 Метод різницевого хешування для первинного виявлення дублікатів	41
2.3.2 Фільтрація розмиття на основі дисперсії оператора Лапласа ...	42
2.3.3 Балансування негативного фону за алгоритмом вибору найвіддаленіших точок.....	43
2.4 Теорія без'якорної детекції YOLO26-nano із вбудованою оптимізацією постобробки (NMS-free)	44
2.4.1 Концепція подвійного призначення міток для детекції без використання NMS	44
2.4.2 Градієнтна стабілізація та оптимізація MuSGD	46
2.4.3 Оптимізація просторових та кольорних перетворень для збереження дрібних цілей	46
2.5 Модифікація фільтрації Калмана для компенсації похибок спостереження за методом OC-SORT	47
2.5.1 Математична формалізація простору станів трекара.....	48
2.5.2 Обчислення просторового вектора неузгодженості.....	48
2.6 Обґрунтування зниження розрядності представлення ваг моделі до формату напівточності (FP16)	49
2.6.1 Теоретичні засади квантування та інтеграція PTQ оптимізації	50
2.7 Обґрунтування джерела отримання польотних даних та специфіки віртуального середовища	52
3 Програмно-алгоритмічна реалізація конвеєра підготовки даних та системи детекції	53
3.1 Програмна реалізація алгоритму KASFraD для декомпресії та мінімізації надлишковості кадрів	53
3.1.1 Збір відеоданих у симуляторі	53
3.1.2 Послідовне декодування відеорядів.....	53

3.1.3	Оцінка зміни кольору за сіткою та гістограмами	54
3.1.4	Оцінка кінематики та збереження результатів	55
3.2	Автоматизація конвеєра розмітки SAM3 із віддаленою синхронізацією даних та верифікацією в CVAT	56
3.3	Розробка програмного модуля паралельної дедуплікації, оцінки дисперсії Лапласіана та вирівнювання фону.....	59
3.3.1	Паралельне обчислення різницевого хешу та оцінка розмиття	59
3.3.2	Видалення дублікатів у межах польотних епізодів.....	60
3.3.3	Балансування фону за алгоритмом FPS.....	61
3.4	Редукція простору класів до категоріального узагальнення «military_object» та фільтрація дрібномасштабних аномалій розмітки ..	62
3.4.1	Об'єднання класів під єдиний ідентифікатор military_object ...	62
3.4.2	Алгоритм очищення вибірки від дрібних обмежувальних рамок	63
3.4.3	Масштабування зображень та запобігання витоку даних	64
3.5	Організація багатопотокового конвеєра апаратного інференсу «YOLO26-nano + OC-SORT» у форматі FP16.....	64
3.5.1	Структура трипотокової асинхронної архітектури	64
3.5.2	Керування чергами та синхронізація потоків	66
3.5.3	Обчислення просторового вектора неузгодженості.....	66
3.5.4	Програмна реалізація конвеєра	68
3.5.5	Оптимізація обчислень у форматі FP16	68
4	Експериментальні дослідження та абляційний аналіз системи	70
4.1	Специфікація експериментального апаратного стенда та структура сформованого датасету.....	70
4.1.1	Специфікація експериментального стенда та вимоги до бортової платформи.....	70
4.1.2	Структурна організація та розподіл кадрів датасету	71
4.2	Оцінка ефективності методів KASFraD та дедуплікації.....	73

4.2.1 Аналіз кількісних показників проріджування та очищення даних	73
4.2.2 Структура та збалансованість сформованого набору даних	75
4.2.3 Теоретичне обґрунтування впливу оптимізації даних на процес навчання	76
4.3 Порівняльний аналіз точності детектора YOLO26-nano у форматах представлення FP32 та FP16	77
4.4 Оцінка стійкості супроводження цілей алгоритмом OC-SORT у разі різкої зміни кутів спостереження та тимчасового перекриття об'єктів ..	79
4.4.1 Оцінка роботи алгоритмів при зміні ракурсу спостереження ..	80
4.4.2 Аналіз стійкості до тимчасового перекриття об'єктів.....	81
4.4.3 Формування керівного сигналу для системи донаведення	82
4.5 Аналіз обчислювальної складності інференсу, затримки та споживання апаратних ресурсів Edge-платформи.....	83
Висновки	85
Перелік джерел посилання	87
Додаток А Програмний модуль кінематично-адаптивного просторово-часового проріджування відеоряду за методом KASFraD.....	94
Додаток Б Програмний модуль оркестрації хмарних репозиторіїв та автоматизованого імпорту завдань у CVAT REST API	95
Додаток В Програмний модуль евристичної дедуплікації зображень та оцінки розмиття.....	96
Додаток Г Програмна конфігурація навчання та валідації без'якорного детектора YOLO26-nano.....	97
Додаток Д Програмний модуль трекінгу цілей та обчислення просторового відхилення	98
Додаток Е Програмний модуль низькорівневої оптимізації та PTQ-компіляції моделі.....	99
Додаток Ж Конфігураційні параметри датасету.....	100

Додаток И Скрипт порівняльної оцінки точності моделей у форматах FP32 та FP16	101
Додаток К Алгоритм розрахунку координат центру та вектора відхилення об'єкта	102
Додаток Л Відомість кваліфікаційної роботи	103

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БПЛА – безпілотний літальний апарат;

РЕБ – радіоелектронна боротьба;

API – Application Programming Interface – інтерфейс прикладного програмування;

CPU – Central Processing Unit – центральний процесор;

CVAT – Computer Vision Annotation Tool – інструмент розмітки комп'ютерного зору;

DFL – Distribution Focal Loss – розподілена фокусна функція втрат;

FPS – Frames Per Second – кадрова частота (кількість кадрів на секунду);

FPV – First-Person View – вид від першої особи;

HOTA – Higher Order Tracking Accuracy – метрика оцінки якості супроводження об'єктів;

HSV – Hue, Saturation, Value – колірна модель «тон, насиченість, значення»;

IoU – Intersection over Union – міра перетину прогнозованої та еталонної обмежувальних рамок;

mAP – Mean Average Precision – метрика середньої точності детекції;

MOT – Multiple Object Tracking – відстеження кількох об'єктів;

NMS – Non-Maximum Suppression – алгоритм придушення немаксимумів;

NPU – Neural Processing Unit – нейропроцесор (обчислювальний співпроцесор);

PTQ – Post-Training Quantization – метод посттренувального квантування моделі;

QAT – Quantization-Aware Training – метод квантування з урахуванням навчання моделі;

SGD – Stochastic Gradient Descent – метод стохастичного градієнтного спуску;

SORT – Simple Online and Realtime Tracking – простий онлайн-трекінг у реальному часі;

SWaP – Size, Weight, and Power – масо-габаритні та енергетичні обмеження системи;

YOLO – You Only Look Once – родина нейромережових архітектур детекції в один прохід.

ВСТУП

Сучасні FPV-дрони є одним із найбільш ефективних засобів ураження наземних об'єктів. Для забезпечення мінімальної затримки передачі відеосигналу в таких системах переважно використовують аналоговий зв'язок, який є критично вразливим до дії тактичних засобів радіоелектронної боротьби (РЕБ). На термінальній ділянці польоту вплив завад або втрата радіогоризонту спричиняють деградацію відеосигналу. Це унеможливорює ручне коригування траєкторії оператором та суттєво знижує ймовірність ураження рухомих цілей [1].

Актуальність роботи зумовлена необхідністю забезпечення високої точності та автономності функціонування FPV-дронів в умовах активної радіоелектронної протидії. Перспективним шляхом вирішення цієї проблеми є перенесення обчислювальних процесів комп'ютерного зору безпосередньо на борт апарата за концепцією Edge AI. Створення автономних програмних модулів супроводження цілей для малопотужних бортових обчислювачів дозволяє утримувати об'єкт у полі зору камери навіть після повної втрати зв'язку з наземною станцією керування [2].

Метою роботи є розробка та оптимізація програмного комплексу виявлення та програмного захоплення цілей у реальному часі для утримання об'єкта бортовими обчислювальними системами БпЛА в умовах обмежених ресурсів.

Для досягнення мети необхідно вирішити такі задачі:

- проаналізувати специфіку відеоданих курсових камер FPV-дронів та сформулювати технічні вимоги до систем комп'ютерного зору;
- обґрунтувати вибір легковагової архітектури YOLO та алгоритму трекінгу для стабільного супроводження об'єктів;
- дослідити методи апаратної оптимізації моделей та зниження розрядності представлення ваг (FP16) для забезпечення високопродуктивного інференсу на Edge-платформах;

- розробити методику напівавтоматизованої підготовки та очищення спеціалізованого набору даних;

- реалізувати інтегрований програмний конвеєр детекції та трекінгу з обчисленням координат відхилення цілі в реальному часі.

Об'єктом дослідження є процес автоматичного розпізнавання та супроводження візуальних об'єктів у відеопотоці в режимі реального часу.

Предметом дослідження є архітектури нейронних мереж YOLO, алгоритми візуального трекінгу, методи їх оптимізації та зниження розрядності представлення ваг для Edge-платформ.

В основу дослідження покладено методи машинного навчання, комп'ютерного зору та математичні алгоритми кінематичного передбачення (фільтрація Калмана). Для оцінки точності та швидкодії системи застосовано метрики mAP, IoU, NOTA, IDF1 та показник кадрової частоти (FPS) із вимірюванням затримки інференсу.

Практична значущість роботи полягає у можливості інтеграції розробленого програмного комплексу в бортові мікрокомп'ютери БПЛА як автономного модуля захоплення цілей. Це дозволяє підвищити точність ураження об'єктів на фінальній ділянці польоту та нівелювати залежність системи від стабільності радіоканалу в умовах активної радіоелектронної протидії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Специфіка предметної галузі (FPV-пілотування)

Ключовою особливістю БпЛА типу FPV є жорстке кріплення курсової камери без використання стабілізаційних підвісів [3]. Для компенсації нахилу корпусу під час руху камера встановлюється під вертикальним кутом від 15° до 30° для маневрування та від 45° до 60° для швидкісних польотів [4]. Орієнтація дрона безпосередньо визначає ракурс зображення, що спричиняє постійну зміну лінії горизонту та геометрії сцени. Відсутність апаратної стабілізації ускладнює роботу алгоритмів автоматичної детекції та трекінгу об'єктів (рисунок 1.1).



Рисунок 1.1 – Характерні аналогові артефакти: зернистий шум, горизонтальні або діагональні лінії та загальне спотворення кадрів

Для забезпечення затримки передавання відеоданих до 20 мс керування FPV-дронами здійснюється через аналогові канали на частоті 5,8 ГГц [2], [5]. Відеопотік піддається впливу завад, що спричиняє

деградацію зображення: зернистий шум, лінії та спотворення кадрів (рисунок 1.1) [5]. Класичні моделі комп'ютерного зору (YOLO), навчені на чистих наборах даних, демонструють падіння точності виявлення під час обробки зашумленого відео через доменний зсув якості зображення [6].

Переміщення на високій швидкості та різкі маневри спричиняють значне розмиття в русі на кадрах [7]. Під час термінального пікірування масштаб цілі експоненціально змінюється. На початкових етапах техніка фіксується як малорозмірна ціль (від 10 пікселів до 50 пікселів). З наближенням її розмір стрімко зростає. Алгоритм супроводження має бути стійким до просторових і масштабних змін та зберігати ідентифікатор цілі без втрати захоплення.

На термінальній стадії під час зниження апарата на висоту від 10 м до 20 м керування ускладнюється. Аналоговий відеосигнал втрачається через перекриття радіогоризонту рельєфом. Додатковим фактором є застосування систем тактичної РЕБ, які блокують відеоканал і телеметрію над ціллю [1]. Втрата зв'язку унеможливорює ручне коригування траєкторії. Розв'язанням проблеми є створення автономної бортової системи машинного зору, яка самостійно утримує ціль і формує керівні вектори на Edge-платформі без участі оператора.

Отже, специфіка FPV-пілотування формує жорсткі вимоги до систем комп'ютерного зору: алгоритми повинні бути стійкими до цифрових і аналогових шумів, різких змін ракурсу і масштабу об'єктів, а також працювати в умовах нестабільного відеозв'язку.

1.2 Порівняльний аналіз наявних систем автоматичного захоплення та супроводження цілей

Практичне застосування систем автоматичного трекінгу цілей залежить від специфіки носія та умов експлуатації. Наявні рішення

поділяються на комерційні комплекси цивільного сектору та спеціалізовані військові системи термінального наведення. Це дозволяє виявити недоліки кожної групи у контексті вимог до систем донаведення ударних FPV-дронів.

Цивільні комерційні системи (ActiveTrack від DJI, алгоритми Skydio) демонструють високу точність розпізнавання об'єктів за ідеальних умов. Проте вони критично залежать від механічної стабілізації та високого співвідношення сигнал/шум. За умов швидкого руху або зміни масштабу об'єкта алгоритм дає збій [8, с. 46]. У разі високої динаміки FPV-дрона та наявності аналогових перешкод їхня ефективність деградує. Пропріетарні закриті архітектури унеможливають їхню інтеграцію у рішення на базі COTS-компонентів.

Військові системи термінального наведення (Switchblade 300/600) оснащені апаратними модулями автоматичного захоплення цілі (ATR). Вони забезпечують надійний супровід цілі на фінальній ділянці, проте є економічно нерентабельними для масового розгортання. Вартість Switchblade 300 становить близько 6000 доларів, а потужніших комплексів перевищує 80000 доларів [9]. Порівняно з FPV-дронами, ціна яких не перевищує 400 доларів, застосування закритих ATR-комплексів не масштабується. Класичні системи стикаються з обмеженнями SWaP-C, що робить їх дорогими для широкого впровадження [10].

Активно розвиваються відкриті defense-tech ініціативи зі створення систем автододоведення для FPV-дронів. Вони використовують архітектури згорткових нейромереж (YOLO) на мікрообчислювачах з NPU. Проблема таких рішень полягає у використанні неоптимізованих нейромереж на слабкому апаратному забезпеченні. Використання базових моделей без квантування спричиняє падіння частоти кадрів. Формується критична латентність контуру керування. Через затримку обчислень генерація керівних векторів відбувається із запізненням, тому дрон минає ціль до завершення циклу системи машинного зору [8, с. 45].

Отже, комерційні цивільні системи не здатні ефективно функціонувати в умовах високої динаміки та завад FPV-пілотування. Військові комплекси є недоцільними для масового використання на базі COTS-компонентів. Відкриті ініціативи стикаються з проблемою критичної латентності на бюджетних обчислювачах. У такий спосіб формується потреба розробки оптимізованої гібридної системи автоматичного захоплення цілі, що забезпечить баланс між Edge AI інференсом у реальному часі та високою точністю на платформах з обмеженими ресурсами. Технічні вимоги до такої системи формалізовано у наступному підрозділі.

1.3 Системні обмеження Edge-платформ та вимоги до обробки даних у реальному часі (SWaP-C)

Розробка завадостійкого алгоритму автономного наведення для сучасних безпілотних літальних апаратів потребує чіткого визначення апаратних умов його функціонування. Оскільки зовнішні канали зв'язку є вразливими до засобів радіоелектронної боротьби (РЕБ), обчислювальний контур має працювати повністю автономно безпосередньо на борту БпЛА [11, с. 12]. Це накладає жорсткі масо-габаритні та енергетичні обмеження (SWaP-C). Для збереження оптимальних льотно-технічних характеристик носія маса обчислювального модуля разом із системою охолодження має становити від 50 г до 100 г, а його пікове енергоспоживання – від 5 Вт до 15 Вт. Компактні розміри платформи та ліміти корисного навантаження зумовлюють дефіцит бортового енергоресурсу [11, с. 3]. Типовим прикладом апаратних рішень, що задовольняють такі вимоги, є доступні ARM-платформи (зокрема Orange Pi 5 Max), вага яких становить близько 62 г за енергоспоживання до 7,3 Вт [12]. Недотримання зазначених лімітів призводить до зменшення маси корисного навантаження, скорочення радіуса дії БпЛА [13, с. 4, 5], а

також до ризику температурного тротлінгу в замкненому корпусі, що провокує алгоритмічні затримки та втрату стабільності системи наведення [13, с. 6].

Робота в межах обмеженого бюджету ваги та енергії безпосередньо визначає межі можливостей обробки даних у реальному часі. Під час термінального пікірування ударні FPV-дрони функціонують в умовах екстремальних навантажень, розвиваючи швидкість понад 100 км/год та зазнаючи значних прискорень [14, с. 1]. Запізнення в обробці візуального потоку навіть на 100 мс є еквівалентним майже 3 метрам сліпого польоту, через що апарат минає ціль до завершення обчислень. Для забезпечення точного керування наскрізна сенсомоторна затримка контуру керування має утримуватися на рівні 40–50 мс [14, с. 4], що формує вимогу до частоти обробки відеопотоку не менше 30 FPS [14, с. 3, 4]. Практичні вимірювання показують, що на бортових Edge-платформах базового рівня час інференсу нейромережі має становити близько 40 мс, залишаючи приблизно 8 мс на обчислення керівних векторів польотним контролером [14, с. 9, 10].

Затримка обробки близько 33 мс дозволяє підтримувати базову пропускну здатність у 30 FPS [14, с. 3]. Водночас агресивне пілотування спричиняє розмиття зображення, втрату візуальних ознак та дрейф координат [14, с. 8], а зниження частоти кадрів генерує міжкадрову невизначеність [14, с. 4, 9], за якої класичні алгоритми супроводження руйнуються. Для збереження безперервності трекінгу результати роботи нейромережі доцільно інтегрувати з показаннями інерціальних датчиків (із частотою від 100 Гц до 200 Гц) за допомогою фільтра Калмана [14, с. 3, 9].

Обчислювальні можливості доступного бортового апаратного забезпечення суттєво обмежують вибір алгоритмів. Класичні мікрокомп'ютери на базі ARM-архітектури без спеціалізованих співпроцесорів під час інференсу забезпечують продуктивність лише від 2 FPS до 5 FPS, що унеможлиблює їх використання в контурі реального часу. Альтернативою є платформи з інтегрованими

нейропроцесорами (NPU) або тензорними прискорювачами (TPU/VPU). Сучасні рішення, такі як Orange Pi 5 з NPU продуктивністю 6 TOPS [12], модулі Google Coral Edge TPU [15, с. 2] чи прискорювачі Hailo-8 (які досягають 26 TOPS за енергоспоживання від 2,4 Вт до 3,3 Вт) [16, с. 7, 13], дозволяють забезпечити необхідну швидкість інференсу понад 30 FPS [17, с. 15, 16].

Проектування системи наведення для таких обчислювачів базується на принципах інженерного компромісу [18, с. 1]. В умовах апаратних лімітів пріоритетом є висока пропускна здатність, що переважає над максимальною точністю розпізнавання (mAP). Легковагові детектори утворюють математичний фронт Парето, де задля швидкодії свідомо знижується метрика mAP [18, с. 18]. У парадигмі Edge AI підвищення швидкості інференсу спричиняє деградацію точності локалізації [18, с. 11], тому ключовою вимогою стає просторова стабільність обмежувальної рамки спільно з жорсткою мінімізацією хибних спрацьовувань. Систематична генерація хибних локалізацій призводить до виходу об'єкта з поля зору сенсора [19, с. 10], тоді як безперервне супроводження запобігає перемиканню алгоритму на фонові завади [19, с. 11]. Окрім того, просторова стабільність геометрії рамки є критичною для правильного обчислення дистанції до цілі [19, с. 22].

Для усунення обмежень пропускної здатності пам'яті та зниження енергоспоживання розгортання моделей високої розрядності на Edge-платформах вимагає застосування методів програмно-апаратної оптимізації. Обмежений ресурс обчислювачів зумовлює необхідність зниження розрядності представлення ваг (квантування) та використання архітектур, адаптованих до цілочисельних операцій [20, с. 7]. Це дозволяє суттєво скоротити затримки пам'яті та прискорити час обробки кадру [21]. Застосування оптимізованих легковагових архітектур, стійких до квантування, є необхідним інженерним кроком для утримання стабільності системи наведення на слабкій бортовій апаратній частині [22].

Кінцевим етапом підготовки є адаптація обчислювального графа під апаратні рушії конкретного мікрообчислювача. Перехід до стандартизованих форматів представлення моделей та їх подальша низькорівнева компіляція дозволяють максимально ефективно утилізувати ресурси апаратних прискорювачів і мінімізувати затримки передачі даних у контурі керування БпЛА.

Таким чином, оптимальним архітектурним рішенням для автономної бортової системи є імплементація легковагових моделей (зокрема NMS-free моделей родини YOLO) у поєднанні з процедурою квантування моделей та алгоритмами візуального трекінгу. Запропонований підхід дозволяє утримувати наскрізну затримку в межах 40–50 мс при частоті обробки понад 30 FPS. Це задовольняє жорсткі обмеження SWaP сучасних БпЛА та формує апаратне підґрунтя для практичної реалізації завадостійкої системи термінального донаведення.

1.4 Дослідження методів детекції та алгоритмів супроводження об'єктів в умовах доменного зсуву даних

Для задоволення технічних вимог до систем реального часу (обробка відеопотоку від 30 FPS, мінімізація латентності інференсу) необхідний ретельний вибір інструментарію. У розділі проаналізовано архітектури неймереж для детекції об'єктів, досліджено алгоритми візуального трекінгу та розглянуто підходи до апаратної оптимізації і квантування моделей для розгортання на Edge-пристроях з огляду на обмеження SWaP.

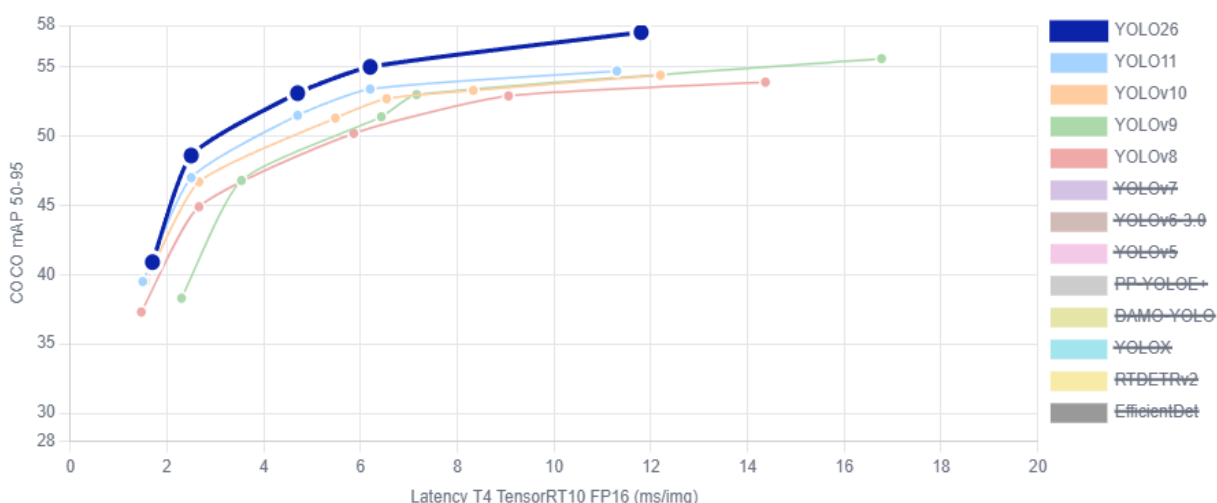
1.4.1 Дослідження методів комп'ютерного зору для детекції (огляд родин моделей, вибір на користь YOLO-подібних)

Задача комп'ютерного зору поділяється на семантичну класифікацію та просторову локалізацію об'єкта в кадрі. Для автономного функціонування

FPV-дрона на термінальній ділянці критично важливою є побудова обмежувальної рамки у реальному часі. Алгоритм детекції має оптимізувати просторову локалізацію та регресію координат рамки, свідомо жертвуючи глибиною класифікації заради мінімізації часу обчислень.

Двоетапні детектори (R-CNN, Fast R-CNN, Faster R-CNN) генерують просторові гіпотези з їх подальшою класифікацією. Вони досягають високої точності розпізнавання, але характеризуються повільним інференсом [23, с. 16]. Ці методи є вимогливими до обчислювальних ресурсів і не підходять для борту БПЛА в реальному часі [23, с. 18]. Важкі двоетапні моделі (Mask R-CNN) потребують потужного апаратного забезпечення [23, с. 42]. Використання неоптимізованих двоетапних архітектур на мікрокомп'ютерах унеможливорює досягнення частоти 30 FPS [23, с. 39].

Оптимальним рішенням для бортових обчислювачів є застосування однопрохідних детекторів (SSD, RetinaNet, родина YOLO). Їхня алгоритмічна парадигма полягає в наскрізній обробці зображення єдиною згортковою мережею, що одночасно генерує ймовірність класу та просторові координати цілі (рисуюнок 1.2).



Рисуюнок 1.2 – Графік залежності COCO mAP від затримки для сучасних архітектур

Згідно з аналізом (рисунок 1.2), однопрохідні моделі формують межу Парето, задовольняючи вимоги жорсткого реального часу та забезпечуючи максимальну швидкість за прийнятною середньою точністю. Зниження точності виявлення мікрооб'єктів є виправданим компромісом, оскільки швидкодія таких систем зростає кратно.

Серед однопрохідних детекторів увагу зосереджено на актуальних поколіннях родини YOLO (YOLOv8, YOLO11, YOLO26).

Відповідно до аналітичних даних (рисунок 1.3, таблиця 1.1), визначальними перевагами цих архітектур є наявність конфігурацій Nano і Small та відмова від використання якорів, що мінімізує обсяг обчислень під час постобробки. Новітні моделі реалізують наскрізне навчання без алгоритму NMS та виключають складну функцію DFL на користь методів прямої регресії [24, с. 7–8].

Усунення NMS і DFL розв'язує проблему розриву між навчанням та використанням, забезпечує детермінований час інференсу та підвищує швидкість обробки на процесорах до 43 % порівняно з попередніми поколіннями [24, с. 7]. Оптимізована архітектура інференсу та підтримка форматів ONNX і TensorRT гарантують ефективну апаратну акселерацію на Edge-пристроях [24, с. 17]. Ці характеристики роблять моделі YOLO оптимальним інструментом для бортових обчислювачів з обмеженнями SWaP (таблиця 1.1).

Отже, базовою архітектурою детекції для бортової системи наведення обрано моделі YOLO у конфігураціях Nano та Small. Таке рішення гарантує баланс між точністю локалізації та обчислювальною ефективністю, дозволяючи досягти високої частоти обробки кадрів. Відмова від NMS-постобробки мінімізує проблему мерехтіння обмежувальних рамок [24, с. 23]. Використання виключно алгоритму детекції є нестабільним підходом, оскільки просторова обробка кожного кадру є ресурсомісткою і спричиняє втрату об'єкта під час різкої зміни ракурсу чи перекриття.

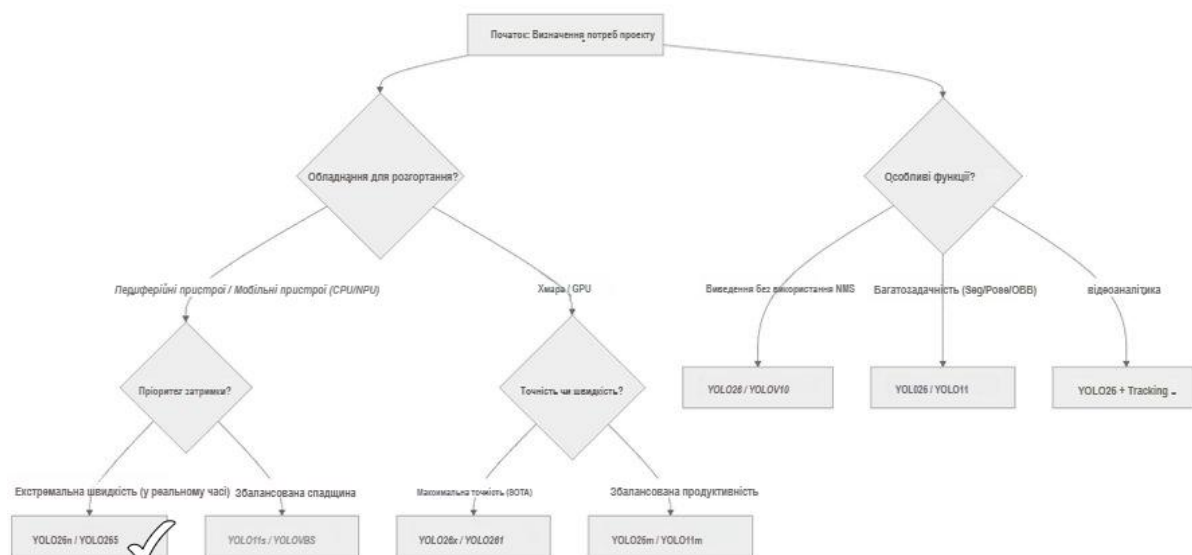


Рисунок 1.3 – Дерево рішень для вибору архітектури на Edge-апаратурі

Таблиця 1.1 – Порівняльні характеристики моделей сімейства YOLO26 на наборі даних COCO

Модель YOLO26	Розмір, пікс	mAP val 50-95 (%)	mAP val 50-95 e2e (%)	Швидкість CPU (мс)	Швидкість T4 (мс)	Кількість параметрів (млн)	Складність FLOPs (млрд)
Nano	640	40,9	40,1	38,9 ± 0,7	1,7 ± 0,0	2,4	5,4
Small	640	48,6	47,8	87,2 ± 0,9	2,5 ± 0,0	9,5	20,7
Medium	640	53,1	52,5	220,0 ± 1,4	4,7 ± 0,1	20,4	68,2
Large	640	55,0	54,4	286,2 ± 2,0	6,2 ± 0,2	24,8	86,4
Extra	640	57,5	56,9	525,8 ± 4,0	11,8 ± 0,2	55,7	193,9

Для безперервного супроводження цілі дискретну детекцію необхідно інтегрувати з математичними алгоритмами візуального трекінгу, що досліджується у наступному підрозділі.

1.4.2 Огляд наявних алгоритмів трекінгу для утримання цілі

Сучасні бортові системи візуального наведення БПЛА функціонують у парадигмі супроводження на основі виявлення. Нейромережевий детектор формує просторові координати об'єктів, які передаються математичному модулю для побудови кінематичних траєкторій [25, с. 2]. Точність

первинної детекції є ключовим фактором багатооб'єктного трекінгу (Multiple Object Tracking, MOT) [26, с. 9]. Покладатися лише на дискретну роботу детектора є неефективним. В авіаційних сценаріях візуальні ознаки стають ненадійними через низьку роздільну здатність сенсорів, зміни масштабу, спотворення та перекриття [25, с. 1, 2]. Це спричиняє флуктуацію рамок, хибне перемикування ідентифікаторів цілей (ID switches) та фрагментацію траєкторій [26, с. 1, 2]. Інтеграція ймовірнісних фільтрів дозволяє здійснювати згладжування траєкторії та передбачення положення цілі, відновлюючи супроводження об'єкта під час збоїв детектора [26, с. 2, 7].

Специфіка застосування ударних дронів передбачає фіксацію скупчень подібних цілей (колон техніки, груп особового складу). Для цього застосовується концепція MOT. Вона вимагає персистентного збереження унікальних ідентифікаторів (ID) для кожного об'єкта. Швидкі маневри БпЛА погіршують прогнозування динаміки об'єктів, спричиняючи втрату ідентифікаторів [25, с. 1]. Механізм просторово-часової асоціації запобігає цим аномаліям. Стійке утримання ID гарантує, що під час маневрів або пікірування система не перемкнеться на фоновий об'єкт.

Базовим алгоритмом MOT є SORT (Simple Online and Realtime Tracking), який комбінує фільтр Калмана для кінематичного передбачення та Угорський алгоритм для асоціації даних. Його еволюцією є DeepSORT, що інтегрує модуль розпізнавання візуальних ознак (ReID) для трекінгу під час перекриття [27, с. 2]. Проте для БпЛА DeepSORT має недолік. Інтеграція ресурсомістких екстракторів ознак (ResNet50) створює неприпустиме обчислювальне навантаження для платформ з обмеженнями [27, с. 7]. Базовий DeepSORT майже втричі повільніший за ByteTrack (31,8 мс проти 11,1 мс). Метрика якості ідентифікації IDF1 становить лише 59,8 % (проти 81,4 % у ByteTrack), а кількість помилок перемикування ідентифікатора є вдвічі більшою [27, с. 11]. Громіздкість моделі

унеможливиює використання DeepSORT у реальному часі на борту дронів [27, с. 11].

Ефективною альтернативою для бортових обчислювачів є алгоритм ByteTrack. Він не використовує ресурсомісткі екстрактори ознак. Метод імплементує дворівневий механізм асоціації, використовуючи рамки з низьким рівнем впевненості. ByteTrack залучає їх для ідентифікації цілей під час зниження точності локалізації [28, с. 1, 2], [29, с. 3]. Алгоритм виокремлює корисну інформацію з детекцій з низькою оцінкою впевненості, що запобігає втраті об'єктів [29, с. 3, 4]. Такий підхід утримує ідентифікатор цілі в умовах деградації візуального сигналу (розмиття в русі), зміни масштабу та перекриття [29, с. 3], [30, с. 4, 5]. Відмова від екстракції ознак гарантує швидкість до 171 FPS. Завдяки оптимізації ByteTrack перевершує SORT і DeepSORT за швидкодією та точністю MOT. Це робить його найбільш раціональним рішенням для інтеграції в Edge-системи реального часу [30, с. 1, 9].

Отож, для забезпечення утримання цілі в умовах завад, різких маневрів БпЛА та скупчення об'єктів доцільним є поєднання швидкісного детектора (YOLO) з алгоритмами множинного трекінгу (ByteTrack). Ця комбінація дозволяє вирішити проблему фрагментації траєкторій, надійно утримувати унікальні ID та здійснювати кінематичне передбачення в реальному часі без перевантаження бортової апаратури. Ефективність такої системи безпосередньо залежить від якості датасету для навчання базового детектора, що аналізується у наступному підрозділі.

1.4.3 Аналіз даних та методів оцінки (особливості FPV-зображень, огляд датасетів, метрики mAP, IoU)

Ефективність нейромережових алгоритмів детекції залежить від домену навчальних даних. Моделі, навчені на звичайних наземних зображеннях, демонструють критичне падіння точності під час обробки

відео з БпЛА через розрив доменів [31, с. 1, 2]. Навчальна вибірка для систем термінального наведення повинна містити динамічне розмиття, артефакти аналогового зв'язку, специфічну авіаційну перспективу, складне тло та малорозмірні цілі [31, с. 2].

Для подолання розриву доменів достатньо локального датасету обсягом від 1500 до 3000 якісно анотованих кадрів за умови застосування методів трансферного навчання з попередньо навченими вагами еталонного набору COCO [32, с. 7, 17]. Використання таких моделей потребує репрезентативності даних, щільної розмітки та адекватних методів аугментації [33]. Без урахування специфічних оптичних артефактів модель втрачає здатність розпізнавати об'єкти в реальному польоті.

Класичні набори даних (DOTA, VisDrone) мають обмежену придатність для навчання алгоритмів тактичних дронів. Набір DOTA сформований на основі супутникових знімків [34, с. 5, 6]. Датасет VisDrone орієнтований на висотну зйомку зі стабілізованих підвісів з ортогональною перспективою [34, с. 4]. Вони суттєво відрізняються від оптичного потоку тактичних FPV-дронів, що функціонують в умовах агресивної просторової динаміки (рисунок 1.4).

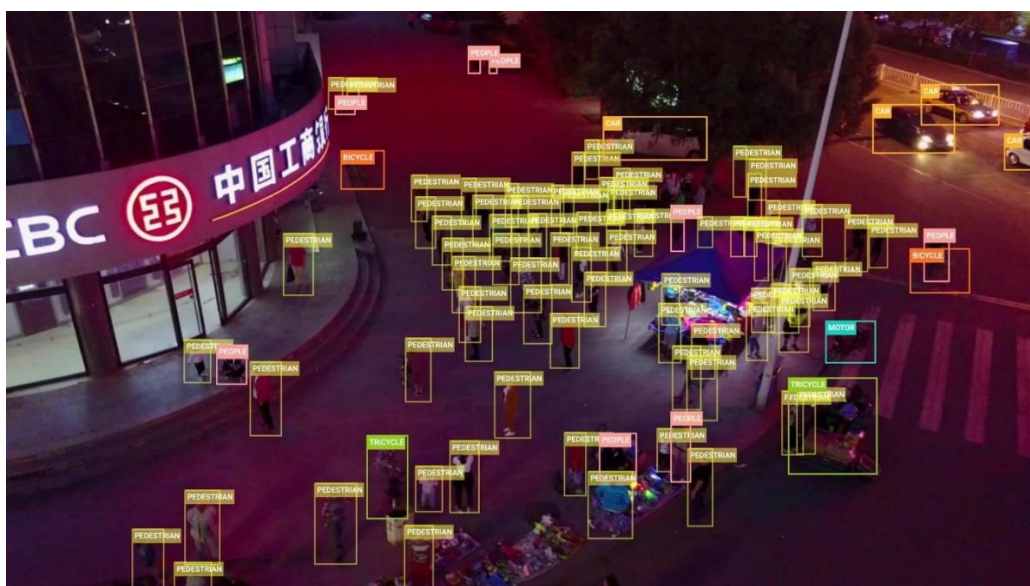


Рисунок 1.4 – Приклад зображень з набору даних VisDrone

Для навчання алгоритмів доцільно використовувати спеціалізовані військові вибірки, такі як Military Vehicles Dataset (рисунок 1.4). Вони містять зображення техніки та фортифікацій з нестационарних ракурсів. Це репрезентує специфіку FPV-середовища: малу висоту, агресивну динаміку та наявність візуальних перешкод. Відкриті масиви даних слід використовувати виключно як базу для попереднього навчання.

Оскільки ідеальних FPV-вибірок немає (рисунок 1.5), застосовують алгоритми трансформації даних. Для легких моделей ефективною є зміна параметрів експозиції (HSV) та просторова трансформація (Mosaic) [35]. Цілеспрямоване введення синтетичних збурень у комбінації з метанавчанням підвищує робастність мережі: точність зростає на 11,7 % за наявності апаратних помилок та на 15,1 % під час візуальних атак [36].



Рисунок 1.5 – Приклад вибірки аеророзвідки в польових умовах

Оцінка якості локалізації цілей здійснюється за допомогою метрики IoU (Intersection over Union). Вона обчислює відношення площі перетину згенерованої рамки та еталонної розмітки до площі їхнього об'єднання. Метрика IoU використовується для асоціації даних та класифікації результату детекції [37], [38].

На базі IoU формується метрика середньої точності mAP. Для автономних БПЛА пріоритетною є базова метрика mAP50, що є достатнім критерієм для оцінки ефективності розпізнавання цілей у реальному часі на Edge-платформах [17, с. 15, 17]. Підвищення порогу IoU спричиняє ефект перенавчання та деградацію точності моделі [39, с. 1, 2]. Такий компроміс у метриках є доцільним: для алгоритмів трекінгу головним є безперервне утримання центру цілі в кадрі. Застосування важких архітектур для надмірної деталізації меж рамки призводить до невиправданих обчислювальних витрат та критичного падіння FPS [17, с. 15].

Отож, проаналізовані класичні набори даних доцільно використовувати виключно для попереднього навчання. Для реалізації парадигми супроводження необхідні спеціалізовані тактичні вибірки обсягом від 1500 до 3000 кадрів із застосуванням методів трансферного навчання. Компенсація дефіциту даних досягається впровадженням алгоритмів аугментації та синтетичних збурень. Використання метрики mAP50 є раціональним інженерним компромісом, оскільки пріоритетом є стабільне утримання центру цілі, а жорсткі вимоги до піксельної точності провокують перенавчання мережі та падіння частоти кадрів. Забезпечення стабільності детекції створює фундаментальне підґрунтя для апаратної оптимізації алгоритмів на бортових Edge-платформах.

1.4.4 Апаратні рішення та методи їх оптимізації (огляд Edge платформ, квантування під слабе апаратне забезпечення)

Інтеграція систем комп'ютерного зору на БПЛА жорстко регламентується масо-габаритними та енергетичними обмеженнями SWaP. Для збереження льотно-технічних характеристик маса обчислювального модуля із системою охолодження має становити від 50 г до 100 г, а пікове енергоспоживання від 5 Вт до 15 Вт. Зразком дотримання цих вимог є

ARM-платформи (Orange Pi 5 Max), маса яких становить близько 62 г, а енергоспоживання не перевищує 7,3 Вт [12].

Недотримання лімітів SWaP спричиняє критичні наслідки для експлуатації БПЛА. Збільшення маси та енергоспоживання обчислювального модуля зменшує масу корисного навантаження та скорочує радіус і тривалість польоту [13, с. 4, 5]. Інтенсивна робота модулів у закритому корпусі без тепловідведення ініціює температурний тротлінг. Це провокує алгоритмічні затримки та втрату стабільності системи наведення у реальному часі [13, с. 6].

Мікрокомп'ютери на базі класичної ARM-архітектури під час інференсу забезпечують продуктивність лише від 2 FPS до 5 FPS, що унеможливорює їх використання у реальному часі. Альтернативою є системи з інтегрованим нейропроцесором (NPU) або тензорними прискорювачами (TPU/VPU). Платформи типу Orange Pi 5 забезпечують продуктивність NPU на рівні 6 TOPS при енергоспоживанні від 5 Вт до 10 Вт [12]. Модулі Google Coral Edge TPU виконують інференс мобільних моделей зі швидкістю до 400 FPS [15, с. 2]. Прискорювачі Nailo8 досягають 26 TOPS при енергоспоживанні від 2,4 Вт до 3,3 Вт [16, с. 7, 13]. Використання тензорних прискорювачів є необхідним для досягнення швидкодії понад 30 FPS [17, с. 15, 16].

Після навчання базові тензори ваг зберігаються у 32-бітному форматі (FP32). Пряме розгортання FP32-архітектур на NPU-платформах є неефективним. Використання повної розрядності генерує вузьке місце пропускної здатності пам'яті. Це зумовлює алгоритмічну надмірність та підвищене енергоспоживання, тому програмно-апаратна оптимізація моделей є обов'язковим етапом.

Фундаментальним методом стиснення є посттренивальне квантування (PTQ) до формату INT8, тобто зниження розрядності ваг від 32-бітних чисел до 8-бітних. Перехід на INT8 зменшує обсяг моделі в середньому вчетверо [20, с. 7]. Завдяки цілочисельній арифметиці на ядрах

NPU швидкість інференсу суттєво зростає. Оптимізація шляхом злиття шарів мінімізує затримки пам'яті, скорочуючи час обробки кадру в понад 2,6 рази [21]. Статистичне падіння середньої точності детекції за умови коректного калібрування є мінімальним, а метод квантування з урахуванням навчання (QAT) зберігає точність на еталонному рівні [20, с. 7]. Новітні NMS-free архітектури (YOLO26) завдяки алгоритмам прямої регресії демонструють високу стабільність під час квантування [22]. Деградація метрик є раціональним компромісом для забезпечення частоти контуру керування на бортових платформах.

Фінальною стадією є інтеграція оптимізованих алгоритмів в апаратне середовище бортового мікрообчислювача. Базовий конвеєр розгортання передбачає експорт моделі у стандартизоване проміжне представлення (ONNX). Наступним кроком здійснюється апаратно-орієнтована компіляція графа обчислень під рушії Edge-платформи. Залежно від архітектури застосовуються компілятори TensorRT, NCNN або RKNN Toolkit. Це дозволяє ефективно утилізувати ресурси кристала та нівелювати програмні затримки.

Отже, оптимальним архітектурним рішенням для бортової системи наведення є імплементація NMS-free моделей родини YOLO, квантованих до формату INT8, у комбінації з алгоритмом візуального трекінгу. Застосування прямої регресії гарантує високу стабільність під час квантування та мінімізує деградацію точності локалізації об'єктів. Запропонований алгоритмічний конвеєр на Edge-платформах із нейроприскорювачами та оптимізований за допомогою компіляції графа обчислень дозволяє нівелювати затримки відеопотоку та забезпечувати частоту інференсу понад 30 FPS. Цей підхід задовольняє жорсткі обмеження SWaP сучасних БпЛА. Напрацьовані принципи підготовки даних, вибору архітектур та методів оптимізації формують підґрунтя для практичної реалізації системи термінального донаведення.

1.5 Постановка науково-практичної задачі та обґрунтування критеріїв оцінки ефективності розробки

Аналіз умов експлуатації безпілотних літальних апаратів (БпЛА) типу FPV свідчить, що на термінальній ділянці польоту аналоговий відеоканал піддається інтенсивному впливу засобів тактичної радіоелектронної боротьби (РЕБ) та ефекту радіогоризонту [1]. Це спричиняє спотворення або повну втрату відеосигналу, що унеможливорює ручне наведення дрона оператором. Для подолання цієї проблеми виникає необхідність автоматизації процесу виявлення та супроводження об'єктів безпосередньо на борту БпЛА [2].

Таким чином, науково-практичною задачею цієї роботи є розробка та оптимізація бортової гібридної системи детекції та візуального трекінгу об'єктів у реальному часі, здатної автономно функціонувати на малопотужних обчислювальних Edge-платформах в умовах завад і тимчасової втрати зв'язку.

Для розв'язання поставленої науково-практичної задачі необхідно вирішити такі підзадачі:

- сформувати спеціалізований набір тактичних навчальних даних, адаптований до низької якості зображення аналогових курсових камер та динамічних умов зйомки (розмиття в русі, перешкоди зв'язку);

- обґрунтувати та адаптувати легковагову архітектуру нейромережевого детектора (класу YOLO) у поєднанні з обчислювально ефективним алгоритмом візуального супроводження (ByteTrack) для надійного утримання цілі за умов зміни її масштабу та часткових перекриттів;

- дослідити та застосувати методи апаратної оптимізації моделей (зокрема квантування ваг) для забезпечення стабільної роботи алгоритмів на малопотужних бортових пристроях з обмеженими ресурсами (SWaP);

– реалізувати програмний конвеєр передачі координат виявленої та зафіксованої цілі до бортового польотного контролера.

1.5.1 Обґрунтування критеріїв оцінки ефективності

Оцінювання ефективності розробленого програмного комплексу детекції та трекінгу об'єктів доцільно здійснювати на основі комплексної системи взаємопов'язаних критеріїв, які охоплюють як точність локалізації цілей, так і загальну обчислювальну продуктивність бортової системи. З урахуванням нормативних вимог до побудови звітів із науково-дослідних робіт, визначено такі ключові критерії оцінки якості розробки:

– середня точність детекції mAP50 (Mean Average Precision при значенні $\text{IoU} = 0.5$), яка визначає здатність моделі правильно локалізувати об'єкт у кадрі; причому за умов апаратної оптимізації допускається помірне зниження цього значення, якщо геометрична стабільність обмежувальної рамки залишається достатньою для коректного обчислення кутових відхилень та траєкторії руху БпЛА;

– метрика перетину множин IoU , яка кількісно визначає точність збігу прогнозованої обмежувальної рамки з еталонною розміткою та використовується для первинної класифікації результатів детекції;

– показники стабільності трекінгу, зокрема кількість втрат цілі та частота перемикання унікальних ідентифікаторів об'єктів (ID-switches), які характеризують стійкість алгоритму до накопичення похибок і його здатність безперервно утримувати цілі в умовах спотворення відеосигналу, перекриття або різких маневрів;

– кадрова частота FPS, яка визначає швидкість обробки відеопотоку на цільовій Edge-платформі; для забезпечення працездатності системи в контурі керування реального часу значення цього показника має бути не меншим ніж 30 FPS;

– затримка інференсу – часовий інтервал, необхідний для повної обробки одного кадру алгоритмом детекції на бортовому обчислювачі; для мінімізації наскрізної сенсомоторної затримки контуру цей параметр не повинен перевищувати діапазон від 40 мс до 50 мс;

– коефіцієнт стиснення та прискорення моделі, що виражає відношення обсягу пам'яті й часу інференсу початкової моделі у форматі FP32 до відповідних параметрів оптимізованої версії; при цьому ефективність апаратної реалізації вважається прийнятною за умови досягнення оптимального використання обчислювальних ресурсів бортового NPU або TPU за мінімальної деградації точності розпізнавання.

Застосування запропонованого комплексу критеріїв забезпечує можливість всебічного й об'єктивного оцінювання відповідності розробленого програмного забезпечення встановленим технічним вимогам, а також підтвердження його готовності до інтеграції в бортові системи керування БпЛА як автономного модуля машинного зору.

2 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ ОПТИМІЗАЦІЇ ДАНИХ ТА АРХІТЕКТУРИ МОДЕЛЕЙ

2.1 Математична модель кінематично-адаптивного просторово-часового проріджування часових рядів за методом KASFraD

Послідовні кадри відеопотоку, отримані з бортової камери безпілотного літального апарату (БпЛА), мають високу часову та просторову надлишковість [23]. Обробка кожного кадру нейромережевими моделями під час навчання є недоцільною через дублювання фонові інформації, яка не несе нового корисного градієнтного сигналу [24], [25]. Для системного вирішення цієї проблеми розроблено математичну модель кінематично-адаптивного просторово-часового проріджування кадрів (Kinematically-Adaptive Spatiotemporal Frame Decimation, KASFraD). Запропонований метод базується на дворівневому аналізі інформативності кадрів, що поєднує оцінювання локальних змін колірно-структурного фону та загальної кінематики руху камери.

На першому рівні обчислювального аналізу оцінюється статичність поточного кадру, що необхідно для виявлення та подальшого вилучення надлишкових сцен у відеопотоці. Цей процес реалізується шляхом розрахунку відстані Бхаттачарія між двовимірними HSV-гістограмами сусідніх кадрів у локальних просторових зонах. Для цього поточний кадр рівномірно розділяється на неперекривні блоки просторової сітки розмірністю $G \times G$ (де $G = 5$), що забезпечує високу чутливість алгоритму до локальних рухів у кадрі. Для кожного отриманого блоку будується нормалізована двовимірна колірна гістограма у каналах тону (Hue) та насиченості (Saturation), що дозволяє нівелювати вплив незначних коливань освітленості. Кількісна математична оцінка близькості розподілу колірних ознак попереднього блоку та поточного визначається за такою формулою відстані Бхаттачарія:

$$d_B(p, q) = \sqrt{1 - \sum_i \sqrt{p(i) \cdot q(i)}}, \quad (2.1)$$

де $d_B(p, q)$ – відстань Бхаттачарія між розподілами кольорних характеристик двох блоків;

$p(i)$ – значення i -го біна гістограми попереднього блоку зображення;

$q(i)$ – значення i -го біна гістограми поточного блоку зображення;

i – індекс відповідного біна двовимірної гістограми.

Як ілюструють наведені на рисунку 2.1 графічні дані, обчислення відстані Бхаттачарія дозволяє з високою точністю й стабільністю оцінити інтегральну міру взаємного перекриття ймовірнісних розподілів у відповідних кольорних каналах.

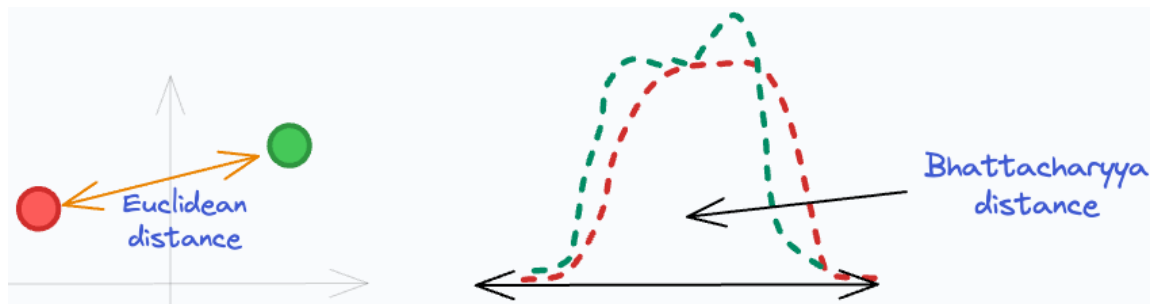


Рисунок 2.1 – Порівняння концепції геометричної евклідової відстані та міри перекриття ймовірнісних розподілів за метрикою Бхаттачарія

Завдяки переходу від вразливого попиксельного аналізу до порівняння статистичних гістограмних характеристик локальних зон, забезпечується підвищена завадостійкість (робастність) розробленого алгоритму до зовнішніх збурень. Це дозволяє ефективно нівелювати негативний вплив миттєвих коливань загальної освітленості сцен, локальних змін яскравості фону, а також природних адитивних шумів та теплових спотворень світлочутливого сенсора відеокамери [24], [26].

Для адаптації чутливості алгоритму до геометрії сцени та специфіки польоту БпЛА локальний поріг децимації визначається динамічно для кожного блоку сітки відповідно до його просторових координат:

$$T_{local}(r, c) = T_{base} \cdot M_{edge} \cdot W(r, c), \quad (2.2)$$

де $T_{local}(r, c)$ – локальний адаптивний поріг децимації для блоку з координатами (r, c) ;

T_{base} – базовий встановлений поріг децимації кадрів;

M_{edge} – коефіцієнт текстурної складності, що визначається на основі щільності меж Кенні [27];

$W(r, c)$ – просторова вага блоку в сітці з координатами (r, c) .

Такий підхід дозволяє штучно знизити чутливість алгоритму у верхній частині кадру, де зазвичай знаходиться лінія горизонту або зона неба, та суттєво підвищити її в центральній фокальній зоні кадру, де очікується поява та подальше утримання потенційного цільового об'єкта.

На наступному, другому ієрархічному рівні функціонування, математична модель KASFrD здійснює комплексний аналіз динаміки зміни масштабів зображення (наближення), а також кутового зближення апарата-носія з цільовим об'єктом. Цей аналіз базується на розрахунку векторного поля розрідженого оптичного потоку за методом Лукаса – Канаде, що дозволяє з високою точністю відстежувати рух локальних геометричних ознак між послідовними кадрами відеопослідовності [28]. З метою компенсації власного руху носія та фільтрації високочастотних шумів, на основі відфільтрованих характерних точок будується стійка афінна модель руху статичного фону за допомогою ітераційного алгоритму RANSAC, який ефективно відсіює викиди та помилкові вектори. Проте в реальних умовах функціонування безпілотного апарата виникають значні просторові деформації та зсуви зображення, викликані вібраціями двигуна та різкими кутовими коливаннями планера. Для нейтралізації цих факторів та

підвищення завадостійкості оцінювання, процес кінематичного аналізу було представлено у вигляді багаторівневої деревоподібної структури, що реалізує алгоритм рекурсивного розгалуження та фільтрації ймовірних кінематичних станів системи.

Ця деревоподібна модель, представлена на рисунку 2.2, ілюструє процес послідовного оцінювання кінематичних змін як фрактального розгалуження на різних ітераціях аналізу часового ряду. Кожен рівень ієрархії відображає глибину дискретизації, де спокійний рівномірний рух веде до мінімального розгалуження, тоді як різкі зміни траєкторії чи наближення до цілі генерують складні каскади станів.

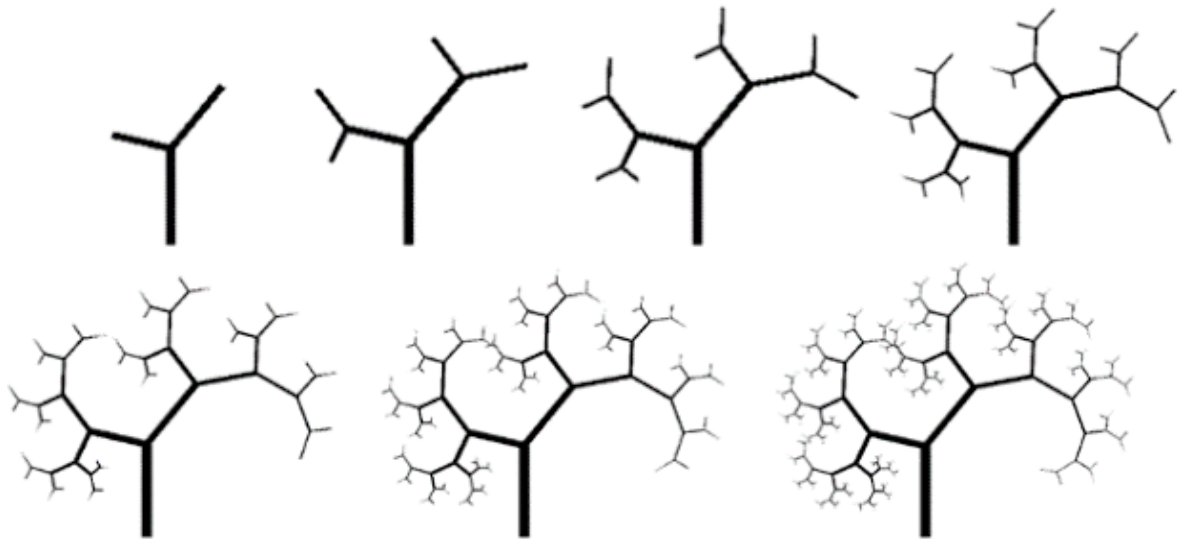


Рисунок 2.2 – Деревоподібна модель мультимасштабного рекурсивного розгалуження та еволюції кінематичних станів відеопотоку

Математично афінне перетворення, що описує динаміку руху фону між суміжними кадрами, визначається парою рівнянь:

$$x' = a_{11}x + a_{12}y + t_x, \quad y' = a_{21}x + a_{22}y + t_y, \quad (2.3)$$

де x', y' – просторові координати характерної точки на поточному кадрі;

x, y – вихідні координати відповідної характерної точки на попередньому кадрі;

a_{ij} – параметри масштабування, обертання та зсуву матриці деформації зображення;

t_x, t_y – параметри трансляції (лінійного зсуву) по відповідних осях координатної сітки.

Для безпосереднього прийняття рішень про інтенсивність проріджування обчислюється коефіцієнт дивергенції (масштабування), який є середнім значенням сингулярних чисел матриці деформації:

$$s = \frac{1}{2} \left(\sqrt{a_{11}^2 + a_{12}^2} + \sqrt{a_{21}^2 + a_{22}^2} \right), \quad (2.4)$$

де s – коефіцієнт масштабування (дивергенції) афінної моделі руху;

$a_{11}, a_{12}, a_{21}, a_{22}$ – коефіцієнти матриці деформації афінного перетворення.

Якщо обчислене значення масштабу перевищує встановлений критичний поріг безпеки (наприклад, $s \geq 1.03$), це свідчить про перехід системи у фазу активного кутового зближення або атаки. У такий момент режим децимації автоматично вимикається, і система переходить до безперервного збереження кожного кадру для збереження динаміки. Програмну реалізацію процедури визначення коефіцієнта масштабування в рамках описаної моделі наведено в додатку А.

Застосування запропонованого гібридного підходу KASFraD дозволяє вилучити до 80 % надлишкових кадрів на статичних ділянках польоту, зберігаючи високу частоту дискретизації в моменти безпосередньої атаки або різких маневрів. Повний програмний код зазначеного модуля інтегровано до загального конвеєра підготовки даних та наведено в додатку А.

2.2 Методологія автоматизованого псевдоанотування (SAM3) та оркестрації хмарних репозиторіїв засобами CVAT REST API

Процес підготовки та розмітки значних масивів візуальних даних для систем комп'ютерного зору традиційно супроводжується високими часовими та трудовими витратами. Для мінімізації ручної праці операторів та прискорення формування навчальних вибірок розроблено конвеєр автоматизованої генерації попередньої розмітки (псевдоанотування). Запропоноване методологічне рішення базується на інтеграції моделі сегментації без попереднього навчання (zero-shot segmentation) Segment Anything Model 3 (SAM3) [29] із хмарною інфраструктурою Hugging Face та інструментарієм верифікації платформи Computer Vision Annotation Tool (CVAT).

Функціонування розробленого інтеграційного конвеєра реалізується через послідовне виконання таких етапів:

- генерація псевдоанотацій: модель SAM3 приймає на вхід проріджені кадри та на основі текстових запитів формує попередні обмежувальні рамки військової техніки та об'єктів інтересу;
- формування та хмарна дистрибуція: отримані кадри спільно з текстовими файлами розмітки автоматично групуються у плоскі ZIP-архіви стандартизованої структури та вивантажуються у хмарне сховище Hugging Face;
- автоматизована оркестрація: спеціалізований скрипт виконує автентифікацію на віддаленому сервері CVAT, ініціює створення нового завдання, завантажує медіафайли та імпортує сформовані псевдоанотації.

Використання інтерфейсу командного рядка CVAT CLI та інструментів REST API дозволяє повністю автоматизувати рутинні операції створення проєктів, виключаючи потребу в ручному налаштуванні через веб-інтерфейс. Ключовий програмний вузол, який відповідає за

ініціалізацію завдання, визначення міток та імпорт локальних архівів у середовище CVAT, наведено в додатку Б.

Після завершення автоматичного завантаження оператор виконує лише швидку верифікацію та точкове коригування меж об'єктів у складних сценах (наприклад, за наявності сильного перекриття або низького контрасту). Фінальний схвалений набір даних автоматично синхронізується з хмарним реєстром Hugging Face, забезпечуючи замкненість циклу підготовки даних. Повна версія інтеграційного інструментарію синхронізації представлена в додатку Б.

2.3 Алгоритмічні засади евристичної дедуплікації зображень на основі різницевого хешування, фільтрації розмиття та методу вибору найвіддаленіших точок (FPS)

Навіть після застосування просторово-часового проріджування відеоряду методом KASFraD у збірному датасеті можуть залишатися приховані дублікати. Вони виникають внаслідок тривалого зависання безпілотного апарату в точці позиціонування або при повторенні ракурсів під час патрулювання. Наявність таких дубльованих кадрів у навчальній вибірці призводить до зміщення ваг нейромережевої моделі та перенавчання на фонові об'єкти. Для вирішення цієї проблеми та забезпечення високої узагальнювальної здатності детектора розроблено алгоритм евристичної дедуплікації та балансування фону.

2.3.1 Метод різницевого хешування для первинного виявлення дублікатів

На першому етапі конвеєра дедуплікації виконується швидке оцінювання візуальної подібності кадрів за допомогою алгоритму різницевого хешування (dHash) [30]. Цей підхід є обчислювально

ефективним і стійким до незначних змін масштабу чи яскравості. Для отримання цифрового відбитка вхідний кадр конвертується у напівтоновий формат та стискається за допомогою методу білінійної інтерполяції до фіксованої роздільної здатності 9×8 пікселів.

Отримане низькорозмірне представлення аналізується шляхом порівняння інтенсивностей суміжних пікселів по рядках. Якщо яскравість поточного пікселя перевищує яскравість його правого сусіда, відповідному біту вектора присвоюється одиничне значення, у протилежному випадку – нульове. Сформований таким чином 64-бітний вектор є унікальним структурним дескриптором зображення. Ступінь подібності між будь-якими двома кадрами оцінюється за допомогою розрахунку відстані Хеммінга, що визначає кількість незбіжних бітів між їхніми дескрипторами. Набір кадрів вважається кандидатом на видалення, якщо відстань Хеммінга між ними не перевищує встановлений експериментальний поріг (у цій роботі поріг прийнято рівним 3).

2.3.2 Фільтрація розмиття на основі дисперсії оператора Лапласа

Для коректного вирішення суперечностей при виявленні групи подібних кадрів (дублікатів) необхідно розробити критерій відбору найбільш інформативного зображення. Використання випадкового вибору є неефективним, оскільки під час руху БпЛА частина кадрів зазнає змазування внаслідок вібрації планера чи різкої зміни кутових швидкостей камери.

Тому при виявленні дублікатів перевага надається кадру з найвищим показником чіткості, а розмиті копії автоматично відсікаються. Оцінювання ступеня фокусування зображення базується на розрахунку відгуку двовимірного диференціального оператора Лапласа, який виділяє області з високими просторовими частотами (контури та межі) [31]:

$$\Delta L = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}, \quad (2.5)$$

де ΔL – відгук оператора Лапласа;

I – інтенсивність яскравості зображення;

x, y – просторові координати пікселів.

На основі отриманої карти меж обчислюється дисперсія розподілу інтенсивностей, яка є інтегральним числовим критерієм чіткості зображення:

$$\sigma^2 = \frac{1}{N} \sum_{x,y} (\Delta L(x, y) - \mu)^2, \quad (2.6)$$

де σ^2 – дисперсія відгуку оператора Лапласа (показник чіткості кадру);

N – загальна кількість пікселів зображення;

$\Delta L(x, y)$ – значення відгуку оператора Лапласа в точці з координатами (x, y) ;

μ – середнє значення відгуку оператора Лапласа по всьому зображенню.

2.3.3 Балансування негативного фону за алгоритмом вибору найвіддаленіших точок

Критичним фактором стабільності навчання глибоких моделей детекції є запобігання зміщенню ваг у бік мажоритарного класу, яким у задачах зйомки з БпЛА є одноманітний порожнє тло (земля, лісові масиви, вода). Для забезпечення балансу класів обсяг негативних кадрів (сцен, що не містять цільових об'єктів розмітки) жорстко лімітується на рівні 10 % від загальної кількості позитивних кадрів.

Простий випадковий відбір цієї підмножини негативних кадрів є неефективним, оскільки він часто призводить до збереження візуально однорідних та просторово близьких ділянок місцевості. Для гарантування

максимального різноманіття відібраного фону застосовано жадібний алгоритм вибору найвіддаленіших точок (Farthest Point Sampling, FPS) [32].

Алгоритм функціонує в просторі 64-бітних dHash-векторів і послідовно обирає кадри, які мають максимальну відстань Хеммінга відносно вже сформованої підмножини. Таке геометричне балансування дозволяє отримати представницьку вибірку фону, що містить 85 % контрастних та чітких сцен різного типу ландшафту та 15 % сцен із помірним розмиттям руху, необхідних для адаптації детектора до динамічних умов інференсу.

Програмну реалізацію алгоритмів обчислення різницевого хешу, розрахунку дисперсії Лапласіана та виділення підмножини кадрів наведено в додатку В.

2.4 Теорія без'якорної детекції YOLO26-nano із вбудованою оптимізацією постобробки (NMS-free)

Для розгортання на бортових обчислювальних модулях БпЛА з жорсткими обмеженнями обчислювальної потужності та енергоспоживання обрано без'якорну (anchor-free) архітектуру детектора YOLO26-nano [33]. Особливістю цієї моделі є повна відмова від класичного алгоритму придушення не-максимумів (Non-Maximum Suppression, NMS) під час безпосереднього інференсу на борту за рахунок впровадження концепції подвійного призначення міток на етапі навчання.

2.4.1 Концепція подвійного призначення міток для детекції без використання NMS

Класичний алгоритм NMS виконує ітераційне сортування та обчислення перекриття рамок кандидатів за метрикою IoU (Intersection over Union). Цей процес традиційно виконується послідовно на

центральному процесорі (CPU), що зумовлює високу латентність обробки зображень та створює «вузьке місце» у реальному часі. У YOLO26-nano цю проблему вирішено шляхом розгалуження головної головки детекції на дві паралельні гілки на етапі навчання:

- гілка багатьох відповідей (One-to-Many): призначає декілька прогнозів для кожного реального об'єкта на основі оптимізатора розподілу завдань (Task-Aligned Assigner) [34]. Це забезпечує сильний градієнтний сигнал для стабільного оновлення ваг згорткових шарів під час зворотного поширення помилки;

- гілка поодиноких відповідей (One-to-One): використовує угорський алгоритм для призначення строго одного найкращого прогнозу на один об'єкт.

Для визначення ступеня відповідності прогнозу реальному об'єкту в гілці багатьох відповідей використовується метрика просторового вирівнювання:

$$t = s^{\alpha} \cdot \text{IoU}^{\beta}, \quad (2.7)$$

де t – коефіцієнт узгодженості прогнозу з реальним об'єктом (Task-Aligned metric);

s – прогнозована ймовірність приналежності об'єкта до відповідного класу;

IoU – міра перекриття між прогнозованою та реальною обмежувальними рамками;

α, β – вагові параметри, що регулюють вплив класифікаційної та локалізаційної складових відповідно.

Під час інференсу на бортовому обчислювачі БпЛА гілка One-to-Many відсікається, а вихідні дані зчитуються виключно з головки One-to-One. Це усуває потребу в обчисленні NMS, знижуючи часові витрати на постобробку до нуля без втрати якості локалізації.

2.4.2 Градієнтна стабілізація та оптимізація MuSGD

Для стабілізації процесу сходження градієнтів у дуальній структурі детектора застосовується спеціалізований оптимізатор MuSGD (Momentum-updated SGD) [35]. На відміну від стандартного стохастичного градієнтного спуску, MuSGD динамічно масштабує величину кроку навчання для кожного параметра залежно від частоти його оновлення в обох гілках детекції.

Це дозволяє нівелювати конфлікти градієнтів, що виникають через суперечливі цілі оптимізації в паралельних гілках One-to-Many та One-to-One на початкових етапах навчання. Як наслідок, забезпечується плавне та швидке сходження цільової функції втрат.

2.4.3 Оптимізація просторових та колірних перетворень для збереження дрібних цілей

Додатковим фактором успішної детекції військової техніки з ракурсу БпЛА є специфічне налаштування просторових перетворень. Експериментально доведено, що стандартні важкі геометричні деформації, такі як випадкове масштабування (scale) чи перспективні зсуви (perspective), призводять до геометричного стиснення дрібних об'єктів. Це викликає деградацію ознак на перших шарах згортки через ефект піддискретизації (downsampling) [36].

Для запобігання цьому явищу в налаштуваннях конвеєра навчання застосовано стратегію збереження роздільної здатності дрібних цілей:

- відмова від масштабування (scale = 0.0): зберігає оригінальну щільність пікселів малих об'єктів та запобігає втраті дрібних деталей;
- вертикальне віддзеркалення (flipud = 0.5): компенсує кутову упередженість зйомки камер БпЛА під різними кутами тангажу;

- поміrne обертання ($\text{degrees} = 15.0$): підвищує стійкість детектора до коливань та різких поворотів носія по крену;
- збільшення діапазону кольорового зсуву ($\text{hsv_s} = 0.3$, $\text{hsv_v} = 0.3$): адаптує модель до різких змін експозиції та наявності глибоких тіней під кронами дерев.

Програмну конфігурацію конвеєра End-to-End навчання детектора YOLO26-nano з урахуванням описаних оптимізацій представлено в додатку Г. Повний програмний код усіх експериментів навчання та валідації наведено в додатку Г.

2.5 Модифікація фільтрації Калмана для компенсації похибок спостереження за методом OC-SORT

Супроводження цілей бортовими камерами БПЛА суттєво ускладнюється через постійні кутові вібрації планера, різкі маневри на термінальній стадії польоту та короточасні перекриття об'єктів перешкодами. Класичний алгоритм SORT у таких умовах часто втрачає ціль, що призводить до накопичення помилок лінійної екстраполяції стандартного фільтра Калмана та різкого зростання кількості перемикачів ідентифікатора траєкторії [37].

Для компенсації похибок спостереження та усунення проблеми накопичення помилок лінійної екстраполяції в модифікованому методі OC-SORT (Observation-Centric SORT) реалізовано такі вдосконалення [38]:

- напрямок з орієнтацією на спостереження, за якого при відновленні контакту з об'єктом після перекриття оцінка швидкості руху у фільтрі Калмана коригується не на основі лінійних прогнозів екстраполяції, а на базі фактичного вектора різниці координат між точкою втрати та точкою повторного виявлення;
- відновлення з орієнтацією на спостереження (Observation-Centric Recovery), яке дозволяє повторно зіставляти втрачені траєкторії зі свіжими

детекціями в локальному часовому вікні, забезпечуючи компенсацію завад від вібрації камери та короткочасного розмиття кадру.

2.5.1 Математична формалізація простору станів трекара

Для моделювання руху цілі у двовимірному просторі зображення вектор стану модифікованого фільтра Калмана описується за допомогою таких параметрів:

$$\mathbf{x} = [x, y, s, r, \dot{x}, \dot{y}, \dot{s}]^T, \quad (2.9)$$

де $\mathbf{x}$ – вектор стану фільтра Калмана;

x, y – просторові координати геометричного центру обмежувальної рамки цілі;

s – площа обмежувальної рамки цілі;

r – відношення сторін обмежувальної рамки цілі;

$\dot{x}, \dot{y}, \dot{s}$ – швидкості зміни відповідних геометричних параметрів стану.

Використання представленого простору станів забезпечує можливість стабільної лінійної апроксимації траєкторії об'єкта на коротких інтервалах часу, коли кутові прискорення камери БПЛА є мінімальними.

2.5.2 Обчислення просторового вектора неузгодженості

Для забезпечення утримання виявленої цілі у фокусі бортової камери необхідно розраховувати керівний вектор неузгодженості. Утримання виконується шляхом визначення просторового відхилення геометричного центру обмежувальної рамки об'єкта від оптичного центру зображення, яке в умовах поточної конфігурації камери має фіксовану роздільну здатність 640×360 пікселів:

$$\Delta X = C_x - S_x, \quad \Delta Y = C_y - S_y, \quad (2.10)$$

де $\Delta X, \Delta Y$ – відхилення геометричного центру цілі від центру зображення по горизонталі та вертикалі відповідно;

C_x, C_y – просторові координати геометричного центру обмежувальної рамки цілі;

S_x, S_y – просторові координати геометричного центру зображення (фокальної точки камери).

Отриманий вектор відхилення передається до виконавчих механізмів стабілізатора камери або польотного контролера для формування коригувальних сигналів керування.

Програмну реалізацію паралельного конвеєра обчислення вектора неузгодженості в реальному часі наведено в додатку Д. Такий підхід гарантує безперервне супроводження цілей навіть під час складних маневрів дрона з різкими кутовими розворотами камери. Повна версія скрипту тестування трекера представлена в додатку Д.

2.6 Обґрунтування зниження розрядності представлення ваг моделі до формату напівточності (FP16)

Бортові обчислювальні платформи функціонують в умовах суворих обмежень щодо тепловиділення та миттєвого споживання енергії. Стандартний формат представлення числових параметрів нейромережі одинарної точності FP32 створює надмірне навантаження на шину пам'яті та обчислювальні конвеєри арифметично-логічних пристроїв бортового процесора. Ефективним інженерним рішенням у цих умовах є перехід до формату напівточності FP16, що дозволяє суттєво скоротити апаратні витрати на зберігання та обробку ваг моделі [39].

Формат FP32 виділяє 1 біт на знак (рисунки 2.3), 8 бітів на експоненту та 23 біти на мантису, забезпечуючи широкий динамічний діапазон і високу

математичну точність. Зі свого боку, формат FP16 використовує лише 16 бітів: 1 біт знаку, 5 бітів експоненти та 10 бітів мантиси.

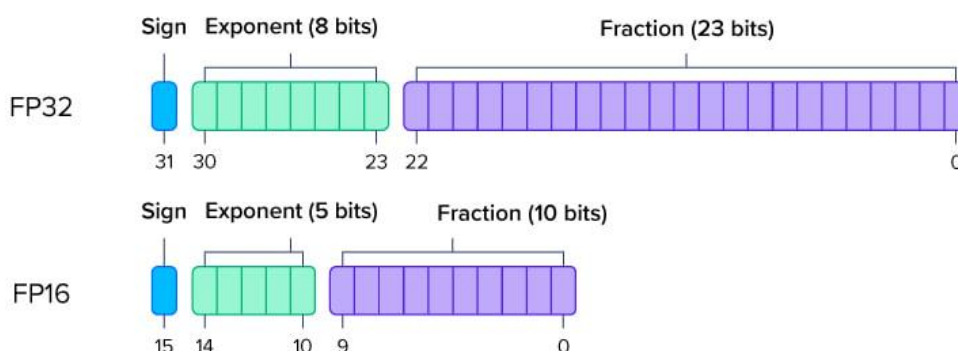


Рисунок 2.3 – Схематичне порівняння розрядності та структури бітів у форматах FP32 та FP16

Така структура дає змогу отримати низку практичних переваг для крайової детекції:

- двократне стиснення обсягу оперативної пам'яті, необхідної для зберігання параметрів моделі, що знижує вимоги до пропускну здатності шини при зчитуванні параметрів згорткових шарів;

- апаратне прискорення обчислень на спеціалізованих тензорних ядрах графічного процесора, оптимізованих для паралельного виконання векторних інструкцій над 16-бітними даними;

- мінімальна деградація точності детекції за метрикою mAP50, що не перевищує десятих часток відсотка порівняно з оригінальним представленням FP32, оскільки діапазону значень мантиси достатньо для збереження репрезентативності ознак згорткових шарів YOLO26-nano.

2.6.1 Теоретичні засади квантування та інтеграція PTQ оптимізації

Для практичного зниження розрядності ваг та оптимізації моделі безпосередньо після завершення навчання застосовується конвеєр

квантування після навчання. Процес переведення дійсних значень ваг у дискретне представлення описується рівнянням рівномірного квантування:

$$q = \text{clip} \left(\text{round} \left(\frac{r}{S} \right) + Z, q_{\min}, q_{\max} \right), \quad (2.11)$$

де q – квантоване цілочисельне значення параметра;

r – вихідне дійсне значення ваги або активації у форматі високої точності;

S – масштабний коефіцієнт квантування (scale factor);

Z – цілочисельна точка зсуву нуля (zero-point);

$q_{\min}, q_{\max}$ – мінімальне та максимальне допустимі цільові значення розрядної сітки.

Розроблений конвеєр PTQ підтримує дві паралельні стратегії оптимізації для крайового обчислювача:

- експорт в апаратно-незалежний формат ONNX FP16, що забезпечує стабільну роботу на широкому спектрі платформ з приведенням усіх числових операторів до напівточності;

- компіляція у формат TensorRT INT8 з автоматичним механізмом FP16 Fallback, що базується на статистичному аналізі активацій на калібрувальному датасеті.

Під час компіляції TensorRT аналізує чутливість кожного шару моделі. Для найбільш критичних згорткових шарів, де квантування до INT8 викликає значну втрату точності або переповнення діапазону, автоматично застосовується механізм відновлення точності до FP16, що гарантує збереження стабільності інференсу [40].

Програмну реалізацію процедури експорту та компіляції оптимізованих PTQ-моделей представлено в додатку Е. Повний вихідний код модуля низькорівневої оптимізації та PTQ-валідації наведено в додатку Е.

2.7 Обґрунтування джерела отримання польотних даних та специфіки віртуального середовища

Навчання моделей детекції в умовах термінального наведення потребує специфічних польотних ракурсів (кутів тангажу й крену), відсутніх у відкритих датасетах типу MS COCO. Тому джерелом первинних відеоматеріалів обґрунтовано обрано віртуальне середовище моделювання. Дані збирали у симуляторі Remote Reaper: FPV Combat Drone Simulator з реалістичною фізикою FPV-квадрокоптера. Накопичення матеріалу здійснювали ручним пілотуванням через реальний радіопульт (USB-інтерфейс), що дозволило детально відтворити автентичні траєкторії польоту. Польоти виконали на трьох локаціях симулятора з різним рельєфом та розташуванням цілей. Для компенсації відсутності динамічної повної погоди варіативність освітлення й фону забезпечили шляхом проведення польотів у різні добові фази: ранкову, денну та вечірню. Записи з курсової камери сегментували на фрагменти за принципом «один запис – одне ураження». Кожен епізод охоплює фази виявлення, пікірування та термінального зближення з технікою, формуючи вихідний масив для подальшої обробки.

3 ПРОГРАМНО-АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ КОНВЕЄРА ПІДГОТОВКИ ДАНИХ ТА СИСТЕМИ ДЕТЕКЦІЇ

3.1 Програмна реалізація алгоритму KASFraD для декомпресії та мінімізації надлишковості кадрів

3.1.1 Збір відеоданих у симуляторі

Первинний набір відеоматеріалів записано під час польотів у симуляторі «Remote Reaper – FPV Combat Drone Simulator» (платформа Steam) [12]. Цей симулятор дозволяє відтворювати динаміку польоту реального безпілотного літального апарата (БПЛА) у повітрі, фізику взаємодії із середовищем, кути нахилу камери при маневруванні та різні типи ландшафтів.

Відеопотік записано у роздільній здатності 1280×720 пікселів із частотою 30 кадрів на секунду. Для спрощення подальшої обробки довгі записи польотів було розділено на короткі кліпи за принципом «один епізод – одне ураження цілі». Окремо збережено фонові відеозаписи, на яких немає військової техніки, але присутні складні маневри над лісами, полями та дорогами.

3.1.2 Послідовне декодування відеорядів

Для читання кадрів із відеофайлів використано спеціалізовану бібліотеку `decord`. На відміну від стандартних засобів `OpenCV`, ця бібліотека забезпечує послідовний доступ до кожного кадру з мінімізацією ризику випадкових зсувів або пропусків у часовому ряді.

У розробленому програмному забезпеченні спочатку здійснюється спроба задіяти графічний процесор за допомогою команди `gpu(0)` для прискорення декодування. Якщо відеокарта не підтримує цей режим або

відсутня в системі, бібліотека автоматично перемикається на роботу з центральним процесором за допомогою команди `cpu(0)`.

3.1.3 Оцінка зміни кольору за сіткою та гістограмами

На першому рівні фільтрації алгоритму KASFraD оцінюється статичність зображення за колірними та структурними ознаками. Кадр переводиться у простір HSV, де вплив коливань освітленості є меншим, ніж у просторі RGB [40]. Після цього зображення ділиться на сітку з 5×5 блоків за допомогою функції `get_block_hists`.

Для кожного з 25 блоків обчислюється двовимірний HSV-гістограма. Порівняння блоків поточного кадру з відповідними блоками попереднього виконується за відстанню Бхаттачарія за допомогою функції `cv2.compareHist` [41].

Процес фільтрації налаштовано під два режими роботи:

- м'який режим, який застосовується до відео з технікою, причому базовий поріг відбору `hist_threshold` встановлено на рівні 0,35, а перевірка кадрів виконується з кроком `stride = 2`;

- жорсткий режим (Hard), який використовується для фонових відеоепізодів, де базовий поріг підвищено до значення 1,0 (діапазон регулювання становить від 0,5 до 2,0), що дозволяє відсіяти значну частину нерухомого ландшафту.

Поріг відбору кадрів адаптується під текстурну складність зображення. За допомогою детектора Кенні (`cv2.Canny`) визначається щільність контурів на кадрі:

- при щільності контурів менше або рівній 0,03 (мало детектованої текстури, наприклад, небо) чутливість підвищується шляхом множення порогу на коефіцієнт 0,5;

- при щільності контурів більше або рівній 0,15 (велика кількість дрібних деталей, таких як трава або гілки) поріг множиться на коефіцієнт 2,0 для уникнення збереження надлишкових кадрів;

- для проміжних значень застосовується лінійна інтерполяція.

Також враховується положення блоків на кадрі через просторові коефіцієнти:

- верхній рядок сітки (небо та лінія горизонту) отримав коефіцієнт 2,5, що знижує чутливість [42];

- бічні стовпчики сітки отримали коефіцієнт 1,8;

- центральна та нижня зони (найбільш імовірні сектори з'явлення цілі) мають коефіцієнт 0,5, що підвищує чутливість;

- нижній рядок сітки отримав коефіцієнт 0,8.

Кадр проходить перший етап фільтрації, якщо відхилення гістограми хоча б одного блоку перевищує його індивідуальний зважений поріг. Окрім цього, у програмному коді реалізовано пошук локальних аномалій: якщо один із блоків змінився сильніше, ніж решта блоків кадру в середньому на три стандартні відхилення (і значення зміни перевищує 0,1), такий кадр вважається важливим і зберігається.

3.1.4 Оцінка кінематики та збереження результатів

На другому рівні фільтрації оцінюється динаміка руху камери. Програма знаходить опорні точки на кадрі за допомогою методу Ши – Томазі та відстежує їх переміщення за допомогою оптичного потоку Лукаса – Канаде (`cv2.calcOpticalFlowPyrLK`).

За знайденими точками через алгоритм RANSAC будується часткова афінна модель руху, що дозволяє оцінити коефіцієнт масштабування (наближення камери до об'єктів) [28]. Програмний код розрахунку цього коефіцієнта наведено у додатку А.

Функція `scale_factor` обчислює зміну масштабу між кадрами. Якщо цей показник перевищує поріг `rapid_scale_on` (1,03), це вказує на швидке наближення БПЛА до цілі. У такому разі вмикається режим прискореного збереження кадрів підряд. Режим вимикається тоді, коли показник масштабування стає меншим за поріг `rapid_scale_off` (1,01).

Для запобігання надто великим пропускам у часі, коли БПЛА зависає на місці або летить над одноманітним ландшафтом, використовується параметр `max_gap_secs` (встановлений на 1,0 секунди). Якщо протягом цього інтервалу жоден кадр не пройшов стандартні фільтри, система виконує примусове збереження поточного кадру.

Кадри, що пройшли відбір, записуються в папку `images` у вихідному форматі 1280 × 720 пікселів. Файли отримують послідовні назви вигляду `frame_0000.jpg`, `frame_0001.jpg` без пропусків в індексах, що дозволяє імпортувати їх до інструментів розмітки або зібрати у відеофайл за допомогою кодека H.264 (AVC1).

3.2 Автоматизація конвеєра розмітки SAM3 із віддаленою синхронізацією даних та верифікацією в CVAT

Створення якісного набору даних для навчання без'якорних однопрохідних методів детекції вимагає значних витрат часу на початковому етапі. Для спрощення цього процесу розроблено двоетапний гібридний конвеєр напівавтоматичного анотування. Його структура поєднує обчислювальні можливості хмарного графічного процесора для генерації попередньої розмітки та локальний сервер розмітки для ручного коригування і перевірки результатів оператором. Проміжне збереження та версіонування всіх матеріалів відбувається у віддаленому хмарному репозиторії.

Для підготовки кадрів, які імпортуються до CVAT, використовується метод кінематично-адаптивного проріджування KASFraD, детально

описаний у підрозділі 3.1. Схема взаємодії між компонентами конвеєра показана на рисунку 3.1.

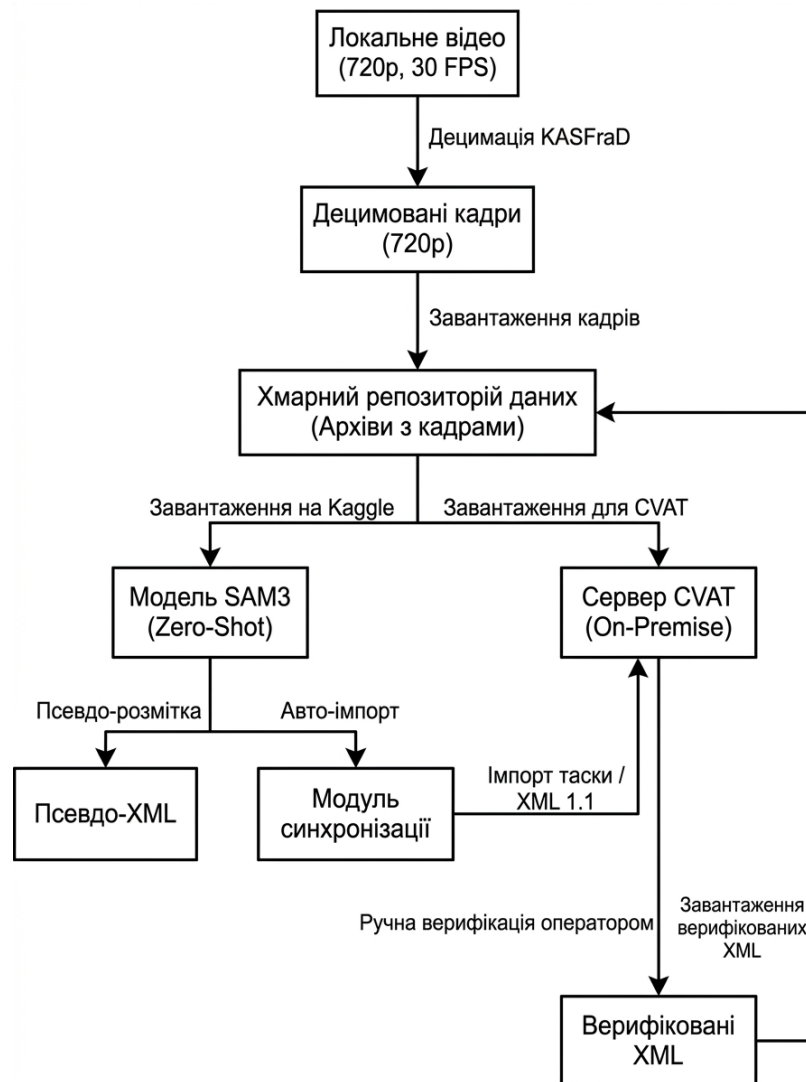


Рисунок 3.1 – Схема взаємодії компонентів під час автоматизованої розмітки

Процес обробки та передачі даних складається з кількох послідовних кроків:

– генерація попередньої розмітки: у хмарному середовищі за допомогою моделі SAM3 та текстових запитів із назвами об'єктів виконується пошук техніки на кадрах із роздільною

здатністю 1280×720 пікселів, а результати роботи моделі записуються як траєкторії руху об'єктів у форматі XML-анотацій [29];

- збереження у хмарі: зображення та відповідна попередня розмітка утворюють єдину структуру папок у віддаленому репозиторії;

- автоматичне створення завдань: програмний модуль синхронізації перевіряє наявність нових даних у хмарі, створює для них завдання на локальному сервері CVAT, налаштовує перелік класів та імпортує кадри разом із попередньою розміткою;

- перевірка та збереження результатів: оператор переглядає розмітку у вебінтерфейсі CVAT, виправляє знайдені помилки, після чого оновлені анотації завантажуються назад у хмару.

Програмний модуль автоматизації взаємодіє з сервером через інтерфейс командного рядка та прикладний програмний інтерфейс (REST API) CVAT. Під час створення нового завдання кадри стискаються у тимчасовий архів без збереження вкладених папок, що зменшує навантаження на мережу під час передачі даних.

Програмний модуль також автоматично обробляє випадки розділення довгих епізодів на менші частини. У такому разі система видаляє застарілі файли з репозиторію та сервера, створює нові завдання для кожної частини окремо та імпортує відповідні анотації. Програмний код, який відповідає за ініціалізацію завдань та імпорт XML-файлів анотацій, наведено в додатку Б.

Після завершення перевірки розмітки оператором програма фіксує стан завдань на сервері. Для цього до REST API CVAT надсилаються запити, які переводять робочі процеси у статус завершених і блокують подальше випадкове редагування. Оновлені XML-файли анотацій зберігаються у хмарному репозиторії в окремому каталозі для верифікованих даних. Використання моделі SAM3 для попередньої розмітки сприяло прискоренню процесу підготовки даних та зменшенню витрат часу оператора порівняно з малюванням обмежувальних рамок вручну.

3.3 Розробка програмного модуля паралельної дедуплікації, оцінки дисперсії Лапласіана та вирівнювання фону

При формуванні навчальних вибірок із відеопотоків, отриманих з камер БпЛА, виникає проблема значного дублювання кадрів. Послідовні кадри відео часто містять майже однакову візуальну інформацію. Це призводить до перенавчання нейромережі та штучного завищення показників точності під час тестування моделі. Водночас надмірна кількість фонових кадрів (зображень без цільових об'єктів) зміщує баланс класів і погіршує здатність детектора виявляти невеликі об'єкти [36].

Первинне проріджування, виконане за допомогою KASFraD (підрозділ 3.1), мінімізує грубу надлишковість, проте для усунення тонких дублікатів та стабілізації фону застосовується окремий програмний модуль попередньої обробки даних.

Щоб зберегти вихідні дані без змін, здійснюється копіювання всього набору даних у нову робочу директорію з подальшою обробкою за таким алгоритмом:

- паралельне обчислення візуальних характеристик та оцінка чіткості для кожного кадру;
- очищення послідовностей від дублікатів у межах кожного польоту з використанням гнучких порогів порівняння;
- відбір найменш схожих фонових кадрів за допомогою алгоритму вибору найвіддаленіших точок для збереження візуального різноманіття.

3.3.1 Паралельне обчислення різницевого хешу та оцінка розмиття

Для швидкої обробки великої кількості кадрів у роздільній здатності 1280×720 пікселів розрахунок параметрів реалізовано у паралельних процесах. Це дозволяє розподілити навантаження на доступні ядра процесора.

Для кожного кадру визначаються два показники:

- різницевий хеш (dHash) розміром 64 біти: для його отримання зображення зменшується до розміру 9×8 пікселів та переводиться у відтінки сірого, після чого порівнюється яскравість сусідніх пікселів [30], причому цей метод дозволяє отримати компактний цифровий опис вмісту кадру, стійкий до змін освітлення;

- дисперсія оператора Лапласіана: цей параметр використовується для визначення чіткості зображення, а якщо значення дисперсії низьке, це свідчить про розмиття кадру через коливання камери або швидкий рух безпілотної [43].

Програмний код для розрахунку цих характеристик наведено у додатку В.

3.3.2 Видалення дублікатів у межах польотних епізодів

Кадри групуються за приналежністю до окремих польотів на основі текстових маркерів у назвах файлів. У межах кожного польоту послідовні кадри порівнюються між собою за допомогою відстані Хеммінга, яка показує кількість розбіжних бітів у їхніх різницевих хешах [44].

Для підвищення точності відбору розроблено систему адаптивних порогів порівняння:

- базовий поріг для кадрів із військовою технікою становить 3, а для фонових кадрів – 14, що сприяє проріджуванню одноманітного фону;

- для польотів над специфічною місцевістю (наприклад, на карті з позначенням «m2») поріг знижується до 1, щоб зберегти унікальні деталі рельєфу;

- для рідкісних класів техніки, таких як гаубиці («howitzer») та бойові машини піхоти («bmr»), встановлено пороги 1 та 2 відповідно, що дозволяє забезпечити збереження кадрів навіть за мінімальних змін ракурсу цілі.

Якщо відстань між сусідніми кадрами менша за встановлений поріг, один із них вважається дублікатом. При цьому аналізується показник чіткості: кадр із меншою дисперсією Лапласіана (більш розмитий) видаляється, а чіткіший залишається в наборі даних. Програмний код реалізації динамічного вибору порогу порівняння наведено у додатку В.

3.3.3 Балансування фону за алгоритмом FPS

Для забезпечення стійкості роботи детектора доцільним є утримання фонових кадрів у кількості 10 % від загального обсягу вибірки (або пропорція 0,1111 від кількості позитивних кадрів). Випадкове видалення фону є менш ефективним, оскільки може призвести до втрати унікальних типів місцевості.

Замість цього застосовано алгоритм вибору найвіддаленіших точок (Farthest Point Sampling, FPS) [32]. Процедура вибору складається з таких кроків:

- попередній відбір чітких кадрів (верхні 70 % від загального фону) та кадрів із природним розмиттям руху (нижні 30 %), причому у фінальну вибірку вони додаються у пропорції 85 % до 15 % відповідно для моделювання реальних умов польоту;

- у межах кожної групи першим обирається найчіткіший кадр, а кожен наступний кадр обирається так, щоб мінімальна відстань Хеммінга від його хешу до хешів уже вибраних кадрів була максимальною.

Це дозволяє зберегти різноманітні фонові пейзажі (ліси, дороги, відкриту місцевість) і уникнути накопичення однотипних кадрів. Програмну реалізацію пошуку найвіддаленіших точок у просторі хешів наведено у додатку В. Повний текст програмної реалізації модуля попередньої обробки перенесено до додатка В.

3.4 Редукція простору класів до категоріального узагальнення «military_object» та фільтрація дрібномасштабних аномалій розмітки

У процесі підготовки навчальної вибірки для бортових систем комп'ютерного зору БПЛА важливим етапом є оптимізація структури даних під обмеження обчислювальної платформи. Наявність великої кількості специфічних класів ускладнює структуру вихідних шарів нейромережевого детектора та підвищує ймовірність помилок класифікації в умовах оптичних завад або дії засобів радіоелектронної боротьби. У цьому підрозділі описано процес переходу до єдиного узагальненого класу `military_object`, а також алгоритм видалення просторових аномалій розмітки.

Після завершення дедуплікації та балансування фону за алгоритмом FPS, принципи якого викладено у підрозділі 3.3, здійснюється уніфікація класів анотацій.

3.4.1 Об'єднання класів під єдиний ідентифікатор `military_object`

Вихідний набір даних містив шість категорій військової техніки: `howitzer` (гаубиця), `antiair` (зенітний комплекс), `bmp` (бойова машина піхоти), `tank` (танк), `cars` (легкові автомобілі) та `truck` (вантажівки). Для задач автоматичного супроводження точна класифікація типу техніки є надлишковою, оскільки головна мета системи полягає у визначенні координат та утриманні цілі в центрі кадру.

Для спрощення вихідного тензора детектора YOLO26-nano всі шість вихідних категорій об'єднано в один клас `military_object` [45]. Таке рішення забезпечує:

- зменшення обчислювальних витрат на вихідних шарах моделі;
- запобігання помилкам класифікації подібних за силуетом об'єктів у складних умовах спостереження (задимлення, маскування, різкі кути огляду).

Технічно це реалізовано шляхом автоматичного перезапису індексу класу в усіх файлах анотацій на значення 0. Процедуру перекодування та уніфікації ідентифікатора класу програмно реалізовано у додатку В.

3.4.2 Алгоритм очищення вибірки від дрібних обмежувальних рамок

Під час зйомки об'єктів на великій відстані виникають надмалі обмежувальні рамки. Наявність таких елементів може ускладнювати якість навчання згорткової моделі з двох причин:

- просторове зменшення у шарах мережі: при послідовному зменшенні масштабу зображення у глибоких шарах нейромережі ознаки дрібних об'єктів втрачаються;

- нестабільність розрахунку похибки: розрахунок функцій втрат локалізації (IoU) для дрібних рамок є менш стабільним, оскільки відхилення прогнозу навіть на один піксель призводить до значних коливань градієнтів під час навчання [46].

Для вирішення цієї проблеми застосовано алгоритм фільтрації рамок за лінійним розміром. Мінімальну межу сторони рамки встановлено на рівні 10 пікселів для зображень розміром 640×360 та 20 пікселів для зображень 1280×720 . Якщо хоча б одна зі сторін об'єкта менша за цей поріг, таку рамку видаляють.

Програмний код перевірки геометричних розмірів обмежувальної рамки наведено у додатку В.

Якщо після очищення дрібних елементів у файлі анотації не залишається жодного об'єкта, програма автоматично видаляє цей файл разом із відповідним зображенням. Це дозволяє уникнути випадкового потрапляння нерозмічених кадрів до тренувальної вибірки як безконтрольного фону.

3.4.3 Масштабування зображень та запобігання витоку даних

Оригінальні відеоматеріали роздільною здатністю 1280×720 пікселів масштабовано до цільового формату детектора – 640×360 пікселів. Зменшення роздільної здатності проведено за допомогою стиснення методом інтерполяції за площею пікселів (Area interpolation). Це сприяє зниженню рівня спотворень меж об'єктів і зберігає чіткість контурів цілей [47]. Для запобігання інформаційному витоку між тренувальним, валідаційним та тестовим наборами даних розділення вибірок проведено на рівні окремих польотних епізодів. Оскільки сусідні кадри одного відеопотоку мають високу схожість, випадкове перемішування кадрів призвело б до штучного перенавчання моделі та отримання недостовірних оцінок точності [48].

Розподіл виконано у співвідношенні 70% (тренувальна частина), 15% (валідаційна частина) та 15% (тестова частина). Фінальний обсяг підготовленого набору даних становить 5735 кадрів, з яких 90% містять об'єкти класу `military_object`, а 10% складають чисті фонові зображення місцевості.

3.5 Організація багатопотокового конвеєра апаратного інференсу «YOLO26-nano + OC-SORT» у форматі FP16

Для реалізації апаратного інференсу використовуються кадри, підготовлені відповідно до процедур масштабування та фільтрації, описаних у підрозділі 3.4.

3.5.1 Структура трипоточної асинхронної архітектури

Для підвищення ефективності бортових систем комп'ютерного зору БПЛА необхідно мінімізувати затримку між реєстрацією зображення

оптичним сенсором та видачею сигналу керування. При послідовній обробці відеопотоку загальний час одного робочого циклу визначається сумою тривалості всіх окремих етапів:

$$T_{\text{cycle}} = T_{\text{capture}} + T_{\text{preprocess}} + T_{\text{inference}} + T_{\text{association}} + T_{\text{dispatch}}, \quad (3.1)$$

де T_{cycle} – загальний час повного робочого циклу, с;

T_{capture} – час захоплення кадру, с;

$T_{\text{preprocess}}$ – час попередньої обробки зображення, с;

$T_{\text{inference}}$ – тривалість роботи детектора, с;

$T_{\text{association}}$ – час роботи алгоритму супроводження, с;

T_{dispatch} – тривалість передачі даних, с.

Будь-які коливання часу обчислень нейромережі безпосередньо впливають на стабільність частоти кадрів у контурі керування БпЛА. Для вирішення цієї проблеми реалізовано асинхронну трипотоківу архітектуру обробки даних на основі стандартної бібліотеки `threading` та системних потокобезпечних черг `queue.Queue` [49]. Розподіл завдань між окремими обчислювальними потоками організовано за такою схемою:

- потік захоплення кадрів (`_frame_capture_thread`): забезпечує безперервне зчитування кадрів з камери із заданою частотою та їх збереження у первинну чергу кадрів;

- потік обробки (`_processing_thread`): виконує підготовку кадру, інференс нейромережі YOLO26-nano та оновлення просторового стану траєкторій в алгоритмі OC-SORT;

- потік відправки результатів (`_output_dispatcher_thread`): здійснює пакування та асинхронну передачу координат цілі до польотного контролера.

3.5.2 Керування чергами та синхронізація потоків

Для стабільної роботи конвєсра в умовах обмежених апаратних ресурсів необхідно виключити накопичення затримки обробки відеоданих. Якщо швидкість надходження кадрів з камери перевищує продуктивність обчислювального блоку, у черзі накопичуються застарілі зображення.

Для запобігання цьому явищу чергу кадрів обмежено місткістю $N = 1$ та застосовано стратегію витіснення застарілих даних [49]. Стан черги Q у момент часу t описується таким виразом:

$$Q_t = \begin{cases} \{F_k\}, & \text{якщо } |Q_{t-\Delta t}| = 0; \\ \{F_k\} \setminus \{F_{k-1}\}, & \text{якщо } |Q_{t-\Delta t}| = 1, \end{cases} \quad (3.2)$$

де Q_t – стан черги у поточний момент часу t ;

F_k – кадр, отриманий від камери в момент часу t ;

$Q_{t-\Delta t}$ – стан черги у попередній момент часу $t - \Delta t$.

Якщо потік обробки зайнятий, потік захоплення кадрів примусово видаляє попередній неактуальний кадр з черги та записує новий. Це обмежує час перебування зображення в черзі величиною, яка не перевищує тривалість одного кадру.

Для розв'язання проблеми затримок при передачі команд керування впроваджено другу чергу результатів. Це дозволяє потоку обробки повертатися до аналізу нового кадру після завершення детекції, тоді як потік відправки результатів транслює дані через фізичні інтерфейси, не затримуючи основний обчислювальний контур.

3.5.3 Обчислення просторового вектора неузгодженості

Кінцевим результатом роботи конвєсра є формування вектора відхилення цілі від точки візування. Роздільну здатність вхідного кадру

після просторового масштабування встановлено на рівні 640×360 пікселів. Геометричний центр зображення визначається як координатна точка відліку:

$$S_x = \frac{W}{2} = 320, S_y = \frac{H}{2} = 180, \quad (3.3)$$

де W – ширина кадру у пікселях;

H – висота кадру у пікселях.

На основі результатів роботи алгоритму OC-SORT для кожної активної траєкторії отримують просторові координати рамки обмеження $[x_1, y_1, x_2, y_2]$. Геометричний центр цілі $[C_x, C_y]$ розраховується за формулами:

$$C_x = x_1 + \frac{x_2 - x_1}{2}, C_y = y_1 + \frac{y_2 - y_1}{2}, \quad (3.4)$$

де x_1, y_1, x_2, y_2 – просторові координати кутів обмежувальної рамки, пікс.

Відхилення координат центру цілі від центра кадру визначає вектор неузгодженості $[\Delta X, \Delta Y]$ у піксельному просторі:

$$\Delta X = C_x - S_x = C_x - 320, \quad \Delta Y = C_y - S_y = C_y - 180, \quad (3.5)$$

де ΔX – відхилення координати центру цілі по горизонталі;

ΔY – відхилення координати центру цілі по вертикалі.

Ці зміщення використовуються бортовим польотним контролером для коригування курсу та тангажу БпЛА з метою утримання цілі по центру кадру камери.

3.5.4 Програмна реалізація конвеєра

Повний вихідний код класу багатопотокової системи детекції та супроводження об'єктів, включаючи розрахунок просторових координат у потоці обробки, наведено у додатку Д.

Для оптимізації роботи детектора поріг впевненості встановлено на рівні $\text{conf} = 0,01$. Це дозволяє виявляти малорозмірні та частково перекриті об'єкти на великих відстанях. Подальше відсіювання помилкових спрацьовувань виконується алгоритмом OC-SORT на основі аналізу історії руху траєкторій.

Використання параметра прямокутного інференсу ($\text{rect} = \text{True}$) знижує потребу в додаванні штучних полів (padding) до кадру, що зменшує витрати обчислювальних ресурсів при підготовці тензорів.

3.5.5 Оптимізація обчислень у форматі FP16

Для підвищення продуктивності конвеєра виконано переведення обчислювальних операцій нейромережі з формату одинарної точності FP32 у формат напівточності FP16 [50]. Перехід до представлення ваг моделі у форматі FP16 забезпечує такі переваги:

- зниження навантаження на шину пам'яті: обсяг даних для збереження коефіцієнтів згорткових шарів та проміжних карт активацій зменшується вдвічі, що підвищує ефективність використання кеш-пам'яті графічного процесора;

- апаратно-залежне прискорення: використання формату FP16 дозволяє задіяти спеціалізовані обчислювальні блоки (Tensor Cores) графічного процесора, які виконують операції матричного множення за меншу кількість тактів;

- оптимізація структури тензорів: при переведенні моделі у формат FP16 активовано структуру розташування даних `channels_last` (формат

пам'яті NHWC), що впорядковує розміщення значень каналів зображення у суміжних комірках пам'яті, підвищуючи ефективність паралельних векторних обчислень.

Завдяки реалізованому комплексу рішень забезпечено підвищення швидкості обробки відеопотоку на бортовому обладнанні при збереженні стабільності формування керівних сигналів.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АБЛЯЦІЙНИЙ АНАЛІЗ СИСТЕМИ

4.1 Специфікація експериментального апаратного стенда та структура сформованого датасету

Проведення експериментальних досліджень розробленої системи потребує визначення характеристик апаратного стенда та опису структури сформованих даних. Це забезпечує можливість повторення експериментів та дозволяє оцінити придатність запропонованих рішень для роботи в умовах обмежених обчислювальних ресурсів.

4.1.1 Специфікація експериментального стенда та вимоги до бортової платформи

Для навчання нейромережових моделей, оптимізації параметрів алгоритмів відстеження та вимірювання часових характеристик системи налаштовано експериментальний стенд. Основним обчислювальним компонентом стенда є графічний процесор NVIDIA T4 з обсягом відеопам'яті 16 GB. Ця апаратна конфігурація забезпечує стабільне середовище для моделювання інференсу та проведення багатопараметричного аналізу.

Експериментальний стенд дозволяє відтворити специфіку низькопоточного оброблення відеоданих та підготувати моделі до подальшої компіляції під цільові апаратні прискорювачі бортової платформи.

Загальні характеристики експериментального стенда та вимоги до цільової платформи наведено в таблиці 4.1.

Таблиця 4.1 – Характеристики моделювального стенда та вимоги до бортової платформи

Компонент системи	Характеристики моделювального стенда	Вимоги до цільової Edge-платформи
Графічний прискорювач	NVIDIA T4 (16 GB GDDR6)	Інтегрований мобільний GPU
Програмне середовище	Linux, PyTorch, CUDA, OpenCV	Linux, апаратний прискорювач інференсу
Основні інструменти	Ultralytics, VoxMOT, TrackEval	Оптимізовані бібліотеки виконання моделей
Цільове призначення	Навчання моделей, профілювання затримок	Автономне супроводження об'єктів у реальному часі

4.1.2 Структурна організація та розподіл кадрів датасету

Вихідні відеоматеріали отримано шляхом реєстрації польотів у симуляторі у роздільній здатності 1280×720 пікселів при частоті 30 кадрів за секунду. На основі цих відеозаписів сформовано послідовні польотні епізоди. Після етапів просторово-часового проріджування, автоматичного маркування та ручної верифікації отримано масив даних об'ємом 5735 кадрів. Моделювання здійснювалося за різних умов освітленості, кутів нахилу оптичної осі камери та висот польоту віртуального носія, що дозволило забезпечити різноманітність візуальних характеристик сцени. Окрему увагу приділено імітації рухомих наземних цілей на складному фоні з наявністю природних та штучних завад. Такий підхід сприяв відтворенню специфіки реального польотного процесу, де кути спостереження постійно змінюються внаслідок маневрування безпілотного апарата. Крім того, до вибірки інтегровано траєкторії зближення та віддалення від об'єктів для забезпечення масштабоінваріантності нейромережевої моделі.

Розподіл датасету на вибірки виконано на рівні цілісних епізодів. Через неподільність польотних епізодів було обрано найбільш близький розподіл до класичного співвідношення 70 / 15 / 15. Такий підхід є важливим для запобігання витоку даних під час навчання, оскільки кадри з

одного відеозапису є висококорельованими й не повинні одночасно перебувати у різних вибірках.

Дійсний розподіл сформованих даних за вибірками наведено в таблиці 4.2.

Таблиця 4.2 – Розподіл епізодів та кадрів за вибірками

Вибірка	Кількість епізодів	Частка епізодів (%)	Кількість кадрів	Частка кадрів (%)
Навчальна (Train)	36	52,9 %	4177	72,8 %
Валідаційна (Val)	15	22,1 %	799	13,9 %
Тестова (Test)	17	25,0 %	759	13,2 %
Всього	68	100,0 %	5735	100,0 %

Для забезпечення стійкості роботи системи в умовах завад та оптимізації процесу навчання застосовано такі методи структурування та попередньої обробки даних:

- редукція простору класів: усі розмічені об'єкти (загальною кількістю 5661 екземпляр) об'єднано в один клас 'military_object', що дозволяє усунути помилки класифікації схожих типів техніки в умовах пилу, туману чи часткового перекриття завадами;

- зниження роздільної здатності: кадри просторово масштабовано до розміру 640×360 пікселів, що зберігає пропорцію кадру 16:9, але у чотири рази зменшує кількість пікселів для обробки, що є критичним для досягнення високої швидкодії на бортовому обчислювачі;

- балансування фону: у загальну структуру вибірок інтегровано 10,0 % негативних (фонових) кадрів (573 кадри), які не містять цільових об'єктів, що запобігає появі помилкових спрацювань детектора на текстурах земної поверхні чи рослинності;

- фільтрація аномалій: з розмітки вилучено обмежувальні рамки, розмір яких менший за 10 пікселів у цільовому форматі детектора, оскільки такі об'єкти не мають інформативних текстурних ознак і створюють нестабільність під час обчислення функцій втрат.

Для інтеграції датасету в середовище навчання використовується конфігураційний файл, що визначає шляхи до папок та назву єдиного класу. Приклад структури цього файлу наведено в додатку Ж.

Програмний код автоматизованого конвеєра підготовки даних, їх дедуплікації та балансування фону перенесено до додатка Ж. На основі сформованого датасету та налаштованого експериментального стенда виконано подальші етапи навчання та дослідження точності й стійкості системи.

4.2 Оцінка ефективності методів KASFraD та дедуплікації

У цьому розділі наведено оцінку ефективності застосування методу кінематично-адаптивного просторово-часового проріджування (KASFraD), а також подальших етапів дедуплікації та балансування даних під час підготовки набору даних версії v25 (з роздільною здатністю 640×360 пікселів). Оцінювання виконано на основі кількісного аналізу структури вибірки на різних етапах обробки, вимірювання ступеня стиснення надлишкових даних та оцінки збалансованості сформованих вибірок.

4.2.1 Аналіз кількісних показників проріджування та очищення даних

Первинний масив відеоданих із віртуального симулятора характеризувався високою надлишковістю та кореляцією суміжних кадрів через стабільність умов польоту. Навчання моделі на таких даних призвело б до передчасного перенавчання та неефективного використання обчислювальних ресурсів графічного процесора.

Застосування методу KASFraD на етапі декодування відео дало змогу вилучити статичні кадри. Шляхом аналізу кольірних гістограм та оптичного

поток метод фіксував лише інформативні фази маневрування та зближення з ціллю, скоротивши обсяг вибірки до 11 585 кадрів (версія v22).

Після вилучення занадто малих обмежувальних рамок об'єктів (менше 10 пікселів) сформовано проміжну версію v23 обсягом 9208 кадрів. На цьому етапі частка фонових кадрів становила 36,5 % (3365 кадрів), що створювало помітний дисбаланс у бік порожнього фону.

Для остаточної оптимізації застосовано модулі дедуплікації, оцінки чіткості за Лапласіаном та балансування фону, що дозволило отримати версію v24 обсягом 5735 кадрів. На її основі шляхом об'єднання класів у категорію 'military_object', ресайзу до 640×360 пікселів та розподілу сформовано фінальну версію v25, кількісні характеристики якої є тотожними версії v24.

Динаміку зміни кількісних та структурних параметрів набору даних на етапах обробки наведено в таблиці 4.3.

Таблиця 4.3 – Динаміка обсягу та структури набору даних на етапах обробки

Параметр вибірки	Початковий датасет (v22)	Після видалення малих рамок (v23)	Оптимізований датасет (v24 / v25)	Загальний відсоток вилучення даних
Загальна кількість кадрів	11 585	9208	5 735	50,5 %
Анотовані кадри (позитивні)	8 220	5 843	5 162	37,2 %
Фоновий клас (негативні)	3 365	3 365	573	83,0 %
Співвідношення (Pos / Neg), %	71,0 / 29,0	63,5 / 36,5	90,0 / 10,0	—

Результати обробки показують, що процедура дедуплікації на основі різницевого хешування (dHash) та відстані Хеммінга дозволила вилучити 681 практично ідентичний анотований кадр, зменшивши обсяг позитивних зображень на 11,7 % порівняно з версією v23. Найбільш вагоме

скорочення відбулося у сегменті фонових кадрів, кількість яких було зменшено з 3365 до 573 зображень (вилучено 83,0 % початкового фону). Завдяки цьому було досягнуто цільового співвідношення: 90,0 % анотованих кадрів та 10,0 % чистого фону.

4.2.2 Структура та збалансованість сформованого набору даних

Після завершення дедуплікації було виконано стратифікований розподіл епізодів на навчальну (Train), валідаційну (Val) та тестову (Test) вибірки за допомогою алгоритму стратифікованого розподілу. Головною умовою під час поділу було збереження цілісності епізодів (усі кадри одного епізоду повинні належати лише до однієї вибірки) для унеможливлення витоку даних.

Фактичний розподіл кадрів та об'єктів у сформованому датасеті версії v25 представлено в таблиці 4.4.

Таблиця 4.4 – Розподіл кадрів та об'єктів за вибірками у датасеті v25

Параметр	Вибірка Train	Вибірка Val	Вибірка Test	Всього
Кількість епізодів	36 (52,9 %)	15 (22,1 %)	17 (25,0 %)	68 (100,0 %)
Кількість кадрів всього	4177 (72,8 %)	799 (13,9 %)	759 (13,2 %)	5735 (100,0 %)
Анотовані кадри	3784 (90,6 %)	714 (89,4 %)	664 (87,5 %)	5162 (90,0 %)
Негативні кадри	393 (9,4 %)	85 (10,6 %)	95 (12,5 %)	573 (10,0 %)
Кількість об'єктів (military_object)	4179 (73,8 %)	734 (13,0 %)	748 (13,2 %)	5661 (100,0 %)

Аналіз даних таблиці 4.4 дозволяє зробити такі висновки щодо якості сформованого набору даних:

– попри жорстке обмеження на неподільність епізодів, алгоритм розподілу забезпечив пропорції вибірок за кількістю кадрів на рівні 72,8 % / 13,9 % / 13,2 %, що є близьким до цільового орієнтиру 70 % / 15 % / 15 %;

- розподіл розмічених екземплярів класу `military_object` відповідає пропорціям вибірок (73,8 % у навчальній, 13,0 % у валідаційній та 13,2 % у тестовій), що запобігає зміщенню оцінки точності детектора під час тестування;

- частка негативних кадрів стабільно утримується в діапазоні від 9,4 % до 12,5 % для всіх трьох підмножин, що гарантує зіткнення моделі з однаковою щільністю фонового шуму під час навчання та валідації.

4.2.3 Теоретичне обґрунтування впливу оптимізації даних на процес навчання

Проведений комплекс заходів із підготовки та структурування даних позитивно вплинув на процес навчання нейромережевого детектора з кількох причин:

- запобігання перенавчанню (*overfitting*), оскільки завдяки вилученню майже ідентичних послідовних кадрів за допомогою різницевого хешування модель втратила можливість простого запам'ятовування конкретних фонових умов або траєкторій, що змусило згорткові шари детектора виділяти більш загальні та інваріантні ознаки військової техніки;

- оптимізація функції втрат, де зменшення частки фонових кадрів до 10,0 % захистило модель від зміщення прогнозування у бік негативного класу (при надмірній кількості порожніх кадрів функція втрат оптимізується переважно за рахунок простого ігнорування об'єктів, що призводить до пропусків цілей під час роботи), а встановлений баланс зберіг чутливість детектора до об'єктів на складному фоні;

- економія обчислювальних ресурсів, оскільки скорочення загального обсягу вибірки вдвічі (з 11585 до 5735 кадрів) без втрати інформативності дозволило пропорційно зменшити час, необхідний для навчання моделі на графічному процесорі, та прискорити збіжність алгоритму оптимізації.

Замість випадкового вилучення кадрів застосовувався алгоритм вибору найвіддаленіших точок (Farthest Point Sampling), програмну реалізацію якого наведено в додатку В.

4.3 Порівняльний аналіз точності детектора YOLO26-nano у форматах представлення FP32 та FP16

Для розгортання моделей глибокого навчання на бортових комп'ютерах безпілотних літальних апаратів потрібно оптимізувати використання обчислювальних ресурсів. Одним із способів прискорення роботи нейромереж є зниження розрядності ваг із формату стандартної точності (FP32) до формату напівточності (FP16). Це дозволяє зменшити обсяг оперативної пам'яті для зберігання параметрів моделі та прискорити виконання арифметичних операцій.

Для уникнення ефекту витоку даних (data leakage) поділ сформованого датасету на навчальну, валідаційну та тестову вибірки здійснювався за цілісними відеоепізодами, а не за окремими кадрами. Оскільки суміжні кадри в межах одного відео є сильно корельованими, їх потрапляння в різні вибірки могло б штучно завищити оцінку точності моделі. У результаті такого поділу тестова вибірка сформувалася з 17 епізодів загальним обсягом 759 кадрів, де представлено об'єкти класу `military_object`.

Важливою особливістю тестового набору є наявність як анотованих кадрів (664 кадри, або 87,5 %), так і чистих фонових кадрів без об'єктів (95 кадрів, або 12,5 %). Це дозволяє об'єктивно оцінити стійкість детектора до хибних спрацьовань у реальних умовах польоту.

Програмний код для запуску порівняльної оцінки на експериментальному стенді з графічним процесором NVIDIA T4 наведено в додатку ІІ.

Під час обох запусків використовували однакові параметри зчитування даних, розмір батчу та апаратне середовище. У другому випадку параметр `half=True` забезпечував приведення ваг моделі та вхідних масивів до типу даних `float16`.

Результати тестування обох конфігурацій детектора наведено в таблиці 4.5.

Таблиця 4.5 – Результати порівняння характеристик детектора YOLO26-nano

Конфігурація моделі	mAP50	mAP50-95	Precision	Recall	Затримка (мс)	FPS
YOLO26-nano (FP32)	0,7587	0,4377	0,8395	0,6845	2,31	433,7
YOLO26-nano (FP16)	0,7563	0,4367	0,8378	0,6845	1,81	552,7

Аналіз отриманих експериментальних даних вказує на такі закономірності:

– зміна показників точності: зниження розрядності ваг призвело до незначного погіршення якості детекції. Значення метрики `mAP50` зменшилося з 0,7587 до 0,7563, що становить лише 0,24 % в абсолютному вимірі. Показник `mAP50-95` знизився на 0,10 % (з 0,4377 до 0,4367). Точність (`Precision`) зазнала незначного спаду на 0,17 % (з 0,8395 до 0,8378). Повнота виявлення (`Recall`) залишилася повністю незмінною та становить 0,6845 для обох конфігурацій. Стабільність показника `Recall` свідчить про те, що перехід до формату `FP16` не спричиняє пропуску об'єктів, які раніше виявлялися базовою моделлю;

– оцінка швидкодії: використання формату `FP16` дозволило зменшити середній час обробки одного кадру. Затримка інференсу зменшилася з 2,31 мс до 1,81 мс, що є швидшим на 21,6 %. Швидкість роботи моделі при цьому зросла з 433,7 FPS до 552,7 FPS (приріст швидкодії становить 27,4 %).

Збереження стабільних метрик точності при переході до 16-бітного представлення пояснюється стійкістю архітектури YOLO26-nano до незначних похибок округлення ваг. Відсутність класичного блоку непересічного придушення знижує кількість послідовних операцій порівняння, де зазвичай накопичуються помилки квантування. Також позитивно вплинув етап очищення вхідних даних, під час якого за допомогою алгоритмів дедуплікації було усунуто надлишкові кадри та сформовано чіткі межі розподілу ознак класу `military_object`.

Таким чином, використання формату FP16 є раціональним рішенням. Воно дозволяє отримати приріст швидкості обробки кадрів за мінімальних втрат точності, що є важливим для бортових систем БпЛА в умовах жорстких апаратних обмежень.

4.4 Оцінка стійкості супроводження цілей алгоритмом OC-SORT у разі різкої зміни кутів спостереження та тимчасового перекриття об'єктів

При використанні бортових систем комп'ютерного зору на безпілотних літальних апаратах (БпЛА) виникають труднощі під час супроводження цілей. Швидкі маневри дрона, зміна кутів спостереження (ракурсу) та тимчасові перекриття об'єктів перешкодами призводять до спотворення форми обмежувальних рамок та втрати траєкторій. Особливу складність становлять сценарії з довготривалими перекриттями, коли об'єкт спостереження зникає з поля зору камери на декілька кадрів під кронами дерев або елементами рельєфу. За таких умов алгоритм відстеження має спиратися на прогнозовану кінематику руху та безпомилково відновлювати унікальний ідентифікатор цілі після її повторної появи на сцені.

Для визначення стійкого рішення проведено експериментальне дослідження на тестовій вибірці, що містить 860 кадрів. Було зіставлено три конфігурації алгоритмів: налаштований ByteTrack, OC-SORT та варіант

OC-SORT із розширеним часом збереження траєкторії та зниженим порогом асоціації (Custom High-Buffer).

Результати оцінювання за допомогою стандартного пакета метрик TrackEval наведено в таблиці 4.6.

Таблиця 4.6 – Порівняльні метрики стійкості супроводження траєкторій на тестовій вибірці

Конфігурація трекара	HOTA, %	DetA, %	AssA, %	IDF1, %	MOTA, %	MOTP, %	IDSW	Frag	MT	ML
ByteTrack (Optimal)	45,69	50,28	41,60	60,98	65,22	72,42	4	4	1	0
OCSORT (Optimal)	52,12	56,48	48,15	65,56	65,94	81,49	4	2	1	0
OCSORT (Custom High-Buffer)	50,12	60,34	41,69	62,10	69,56	81,55	6	2	1	0

4.4.1 Оцінка роботи алгоритмів при зміні ракурсу спостереження

Різка зміна кутів огляду при маневруванні БпЛА призводить до того, що прогнози стандартного фільтра Калмана починають суттєво відхилятися від реального положення цілі. Через це зменшується площа перетину між передбаченою областю та новою детекцією від нейромережі.

Алгоритм ByteTrack у таких умовах часто втрачає об'єкт, що підтверджується найнижчою точністю асоціації траєкторій ($AssA = 41,60\%$) та інтегральною метрикою HOTA на рівні 45,69 %. Це пояснюється його залежністю від точності лінійного прогнозування фільтра Калмана.

Натомість алгоритм OC-SORT використовує модель імпульсу руху (Observation-Centric Momentum). Вона враховує напрямок переміщення об'єкта на основі довшої історії спостережень. Це дозволяє компенсувати помилки прогнозування при різких кутових зсувах камери. Як результат, конфігурація OC-SORT (Optimal) показала вищу стабільність: метрика HOTA зросла до 52,12 %, а точність асоціації ($AssA$) – до 48,15 %.

4.4.2 Аналіз стійкості до тимчасового перекриття об'єктів

У разі часткового або повного перекриття цілі (наприклад, деревами чи елементами рельєфу) детектор тимчасово перестає виявляти об'єкт (рисунок 4.1). У цей момент якість роботи системи залежить від здатності трекера зберігати інформацію про траєкторію в пам'яті (параметр `max_age` або `track_buffer`), наведено на рисунку 4.2.



Рисунок 4.1 – Приклад супроводження цілі в умовах короточасного перекриття та зміни ракурсу спостереження із візуалізацією вектора відхилення $[\Delta X, \Delta Y]$

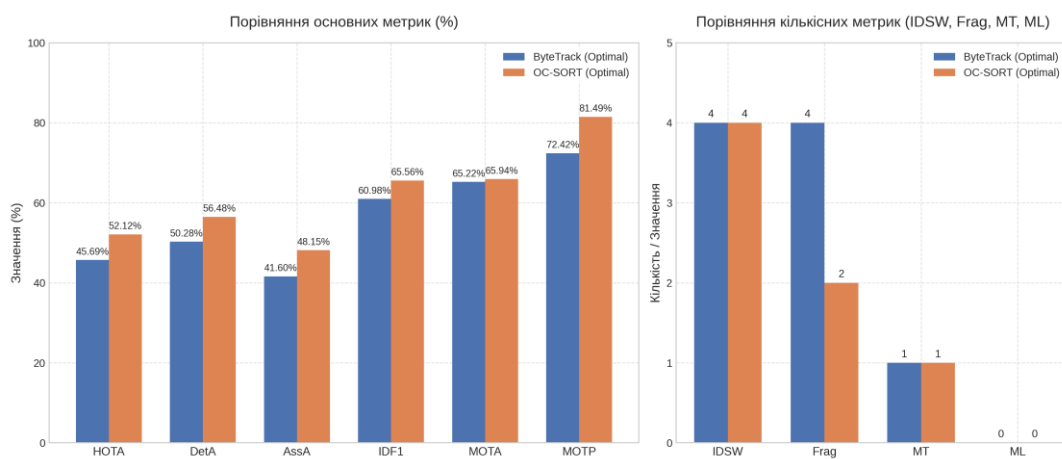


Рисунок 4.2 – Гістограми розподілу метрик стійкості супроводження (HOTA, IDF1, IDSW) для досліджуваних конфігурацій трекерів

Оптимальний варіант OC-SORT продемонстрував меншу схильність до дроблення траєкторій. Кількість фрагментацій (Frag) у нього становить 2, тоді як у ByteTrack цей показник дорівнює 4. Обидва алгоритми при цьому зберегли однакову кількість перемикачів ідентифікаторів ($IDSW = 4$).

При аналізі кастомної конфігурації OC-SORT (Custom High-Buffer), де поріг асоціації було знижено до $iou_threshold = 0,10$, а час пам'яті збільшено до $max_age = 30$ кадрів (близько 1 секунди реального часу), виявлено такі закономірності:

- метрика покриття детекцій (DetA) покращилася до 60,34 %, а МОТА – до 69,56 %, що вказує на те, що система частіше утримує ціль у складних умовах;

- водночас точність асоціації траєкторій (AssA) знизилася до 41,69 %, а кількість перемикачів ідентифікаторів ($IDSW$) зросла до 6.

Зниження порогу асоціації змушує трекер пов'язувати з наявною траєкторією випадкові завади або помилкові детекції під час тривалої відсутності об'єкта. Це наочно демонструє компроміс між повнотою виявлення цілей та точністю збереження їхніх унікальних номерів.

4.4.3 Формування керівного сигналу для системи донаведення

Для автоматичного утримання цілі бортовий обчислювач у реальному часі розраховує вектор неузгодженості між геометричним центром знайденого об'єкта та центром кадру. Фрагмент програмної реалізації цього процесу наведено в додатку К. Повний код інтегрованого модуля детекції та супроводження міститься в додатку Д.

Часті втрати траєкторії (Frag) або зміна ідентифікатора об'єкта ($IDSW$) створюють різкі стрибки значень вектора відхилення (dx , dy). Це призводить до нестабільної роботи приводів камери БпЛА або автопілота загалом.

Завдяки меншій кількості фрагментацій ($\text{Frag} = 2$) та високій точності асоціації ($\text{AssA} = 48,15\%$) алгоритм OC-SORT (Optimal) забезпечує плавну зміну координат керівного вектора. Це дозволяє уникнути різких коливань у контурі керування БПЛА під час тимчасової відсутності візуального контакту з об'єктом.

Таким чином, використання оптимально налаштованого трекера OC-SORT разом із без'якорним детектором YOLO26-nano дозволяє стабільно супроводжувати рухомі наземні об'єкти в умовах завад та інтенсивного маневрування.

4.5 Аналіз обчислювальної складності інференсу, затримки та споживання апаратних ресурсів Edge-платформи

Оцінку обчислювальної складності розробленого конвеєра супроводження цілей виконано в хмарному середовищі Kaggle на базі графічного процесора NVIDIA T4 (з обсягом відеопам'яті 16 ГБ) [31]. Проведення прямих вимірювань або екстраполяції на цільові апаратні платформи малого форм-фактора (Edge AI) без попередньої апаратно-залежної компіляції (RKNN, TensorRT тощо) є нерепрезентативним у межах дослідження. З цієї причини аналіз сфокусовано на оцінці чистої алгоритмічної затримки окремих модулів та ефективності оптимізації форматів представлення даних.

Порівняльний аналіз форматів представлення ваг детектора YOLO26-nano демонструє перевагу використання напівточності. Перехід від формату одинарної точності (FP32) до напівточності (FP16) дозволив знизити затримку інференсу детектора на 21,6 % – з 2,31 мс (433,7 FPS) до 1,81 мс (552,7 FPS). Така оптимізація є важливою, оскільки саме етап детекції є найбільш ресурсомістким етапом (пляшковим горлом) системи. При цьому зниження точності за метрикою mAP50 є незначним та становить

лише 0,24 % (зниження з 0,7587 до 0,7563), що обґрунтовує використання формату FP16 у контурах бортових систем.

Аналіз часових витрат блоку трекінгу показує, що оптимальна конфігурація OC-SORT забезпечує затримку асоціації на рівні 0,951 мс на кадр (1051,2 FPS) [32]. Це лише на 9,4 % повільніше за простіший базовий трекер ByteTrack (0,869 мс / 1150,5 FPS). Враховуючи покращення якості супроводження (зростання інтегральної метрики НОТА на 6,43 % в абсолютному еквіваленті), додаткові обчислювальні витрати у розмірі 0,082 мс на кадр є прийнятними. Загальна наскрізна латентність обробки одного кадру в конвеєрі дорівнює сумі послідовних кроків (інференс YOLO26 FP16 та асоціація OC-SORT) і становить 2,761 мс (1,81 мс та 0,951 мс), що задовольняє жорсткі вимоги до сенсомоторної затримки контуру керування БПЛА (40–50 мс). Завдяки багатопотоковій організації черги обробки (п. 3.5), де детекція та трекінг суміжних кадрів виконуються паралельно, пропускна здатність конвеєра обмежується виключно часом роботи найповільнішої ланки (інференс детектора = 1,81 мс), що дозволяє підтримувати стабільну кадрову частоту обробки на рівні до 552 FPS.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто поставлену мету, у повному обсязі виконано всі сформульовані у вступі задачі та розв'язано науково-практичне завдання з розробки й оптимізації програмного комплексу для автономного виявлення та трекінгу об'єктів у реальному часі.

Завдяки застосуванню алгоритму кінематично-адаптивного проріджування KASFraD вдалося вилучити до 80% надлишкових кадрів на статичних ділянках польоту. Подальше очищення за допомогою різницевого хешування dHash та оцінки чіткості за допомогою дисперсії оператора Лапласа дозволило скоротити обсяг вибірки на 50.5% – до 5735 кадрів, з яких 90% містять цільові об'єкти, а 10% складають чисті фонові зображення. Генерацію попередньої розмітки було автоматизовано шляхом впровадження моделі нульового навчання SAM3 та інструментів оркестрації CVAT REST API, що суттєво зменшило витрати часу на ручне маркування.

Як базовий детектор обґрунтовано та адаптовано без'якорну архітектуру YOLO26-nano. Проведений порівняльний аналіз показав, що зниження розрядності ваг моделі до формату напівточності FP16 є раціональним інженерним рішенням. Це дозволило скоротити затримку інференсу детектора на 21.6% – з 2.31 мс до 1.81 мс, і підвищити швидкість обробки на 27.4% – до 552.7 FPS на тестовому стенді з графічним процесором NVIDIA T4. При цьому зниження точності за метрикою mAP50 є незначним і становить лише 0.24% – з 0.7587 до 0.7563, а повнота виявлення Recall залишилася повністю незмінною на рівні 0.6845, що гарантує відсутність пропусків цілей.

Для стабільного супроводження об'єктів в умовах різких маневрів та тимчасових перекриттів реалізовано модифікований алгоритм OC-SORT. У порівнянні з базовим трекером ByteTrack, обраний метод показав кращу

стійкість: інтегральна метрика НОТА зросла до 52.12%, а точність асоціації AssA склала 48.15% при затримці обробки всього 0.951 мс на кадр. Інтегрований програмний конвеєр детекції та трекінгу забезпечує загальну наскрізну латентність обробки кадру на рівні 2.761 мс, що з запасом задовольняє жорсткі вимоги до сенсомоторної затримки контуру керування безпілотного апарата в межах 40-50 мс. Розрахунок просторового вектора відхилення від геометричного центру кадру дозволяє стабільно передавати керівні сигнали на польотний контролер.

Практична цінність розробки полягає в можливості створення повністю автономного модуля захоплення цілей, здатного функціонувати в умовах втрати зв'язку та активної роботи засобів радіоелектронної боротьби. Спроектований конвеєр орієнтований на апаратне розгортання на малопотужних вбудованих обчислювачах з обмеженням SWaP, де основною цільовою платформою є мікрокомп'ютер NVIDIA Jetson Orin Nano. Це дозволяє забезпечити високу автономність без зниження корисного навантаження або радіуса дії безпілотника.

Створені програмні модулі декомпресії та дедуплікації зображень, а також конфігурації для навчання без'якорних моделей детекції можуть бути використані у навчальному процесі для підготовки студентів за напрямками штучного інтелекту, комп'ютерного зору та розробки систем реального часу.

У межах виконаної роботи експериментальна оцінка швидкодії та оптимізація форматів представлення ваг проводилися на симульованому стенді на базі графічного процесора NVIDIA T4. Тому безпосереднім завданням для майбутніх досліджень є перенесення розробленого програмного контуру на фізичну апаратну платформу NVIDIA Jetson Orin Nano, проведення натурних льотних випробувань, а також дослідження можливості більш глибокого квантування моделі до формату INT8 із застосуванням інструментів TensorRT для подальшого зниження навантаження на обчислювальні ресурси.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Schwennessen P. Ukraine's DIY drones defy Russian jamming – GIS Reports. *GIS Reports*. URL: <https://www.gisreportsonline.com/r/ukraine-diy-drones> (date of access: 15.05.2026).
2. FPV, квадрокоптери, рої: як працюють технології сучасної війни? *VGI-9*. URL: <https://vgi.com.ua/fpv-kvadrokopty-ty-royi-yak-suchasni-drony-zminyuyut-vijnu-ta-tehnologiyi/> (дата звернення: 16.05.2026).
3. Складові FPV дрона та основні параметри. *Victory-drones*. URL: <https://pub-d029182fe93742ef990512aafa639433.r2.dev/FPV%20posters/Складові%20FPV%20дрона%20та%20їхні%20параметри.pdf> (дата звернення: 16.05.2026).
4. Який кут нахилу камери найкращий для FPV-дрона? *fpv-manuals*. URL: https://fpv-manuals.victory-drones.com/docs/Introduction_to_FPV/Camera_angle/ (дата звернення: 16.05.2026).
5. Liang O. How to fix analog FPV video issues: noise, scrolling, power, signal, and compatibility. *OscarLiang*. URL: <https://oscarliang.com/connect-vtx-fpv-camera/> (date of access: 16.05.2026).
6. Deshmukh S., Hambarde K., Proença H. Generalization under scrutiny: cross-domain detection progresses, pitfalls, and persistent challenges. *arXiv.org*. URL: <https://doi.org/10.48550/arXiv.2604.08230> (date of access: 16.05.2026).
7. Drone detection with event cameras / G. Magrini et al. *2025 IEEE/CVF international conference on computer vision workshops (ICCVW)*, Honolulu, HI, USA, 19–20 October 2025. 2025. P. 4762–4773. URL: <https://doi.org/10.1109/iccvw69036.2025.00494> (date of access: 09.07.2026).
8. Habash N., Alqumsan A. A., Zhou T. Recent real-time aerial object detection approaches, performance, optimization, and efficient design trends for

onboard performance: a survey. *Sensors*. 2025. Vol. 25, no. 24. P. 7563.
URL: <https://doi.org/10.3390/s25247563> (date of access: 09.07.2026).

9. Dean S. E. Loitering munitions: production scaling kicks into higher gear. *European Security & Defence*. URL: <https://euro-sd.com/2025/08/articles/armament/46009/loitering-munitions-production-scaling-kicks-into-higher-gear/> (date of access: 17.05.2026).

10. How could FPV drones change warfare? *The Economist*. URL: <https://www.economist.com/the-economist-explains/2023/08/04/how-could-fpv-drones-change-warfare> (date of access: 17.05.2026).

11. Hashim H. A. Advances in UAV avionics systems architecture, classification and integration: a comprehensive review and future perspectives. *Results in engineering*. 2024. P. 103786.
URL: <https://doi.org/10.1016/j.rineng.2024.103786> (date of access: 09.07.2026).

12. Orange Pi developer portal. *Orange Pi*. URL: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html> (date of access: 18.05.2026).

13. A survey of object detection for uavs based on deep learning / G. Tang et al. *Remote sensing*. 2023. Vol. 16, no. 1. P. 149.
URL: <https://doi.org/10.3390/rs16010149> (date of access: 09.07.2026).

14. Champion-level drone racing using deep reinforcement learning / E. Kaufmann et al. *Nature*. 2023. Vol. 620, no. 7976. P. 982–987.
URL: <https://doi.org/10.1038/s41586-023-06419-4> (date of access: 09.07.2026).

15. Google LLC. Coral – USB accelerator datasheet version 1.0. *CDN reichelt* – 403. URL: <https://cdn-reichelt.de/documents/datenblatt/A300/CORAL-USB-ACCELERATOR-DATASHEET.pdf> (date of access: 18.05.2026).

16. Hailo M.2 AI acceleration module. *Hailo*. URL: <https://neardi.com/uploads/files/20240625/b9e4b178526e1556c713ceaa3ef40c9b.pdf> (date of access: 18.05.2026).

17. Hakani R., Rawat A. Edge computing-driven real-time drone detection using yolov9 and NVIDIA jetson nano. *Drones*. 2024. Vol. 8, no. 11. P. 680. URL: <https://doi.org/10.3390/drones8110680> (date of access: 09.07.2026).
18. Small-Object detection at the edge: A pareto-efficient benchmark of lightweight YOLO models on UAV and overhead datasets / B. Shakyia et al. *IEEE access*. 2025. P. 1. URL: <https://doi.org/10.1109/access.2025.3648968> (date of access: 09.07.2026).
19. Van der Aa C. Design of an image-based visual servoing system for autonomous quadcopter interception. *TU delft repository*. URL: <https://resolver.tudelft.nl/uuid:4d8f5709-c20e-4ed2-9f62-26ce3ecde223> (date of access: 17.05.2026).
20. Soliman Y. A., Ghoneim A., Elkhoully M. Multi-Class object detection using quantized yolov11 for real-time inference. *International journal of advanced computer science and applications*. 2025. Vol. 16, no. 12. URL: <https://doi.org/10.14569/ijacsa.2025.0161283> (date of access: 09.07.2026).
21. TensorRT export for YOLO26 models, ultralytics docs. *Ultralytics Docs*. URL: <https://docs.ultralytics.com/integrations/tensorrt> (date of access: 18.05.2026).
22. YOLO26: the edge-first evolution of real-time object detection | datature blog. *Datature: Computer Vision AI Platform for Enterprises & Developers*. URL: <https://datature.io/blog/yolo26-the-edge-first-evolution-of-real-time-object-detection> (date of access: 18.05.2026).
23. Convolutional Neural Network-Based Block Up-Sampling for HEVC / J. Lin et al. *IEEE transactions on circuits and systems for video technology*. 2019. Vol. 29, no. 12. P. 3701–3715. URL: <https://doi.org/10.1109/tcsvt.2018.2884203> (date of access: 09.07.2026).
24. Low illumination video image enhancement / Z. Li et al. *IEEE photonics journal*. 2020. Vol. 12, no. 4. P. 1–13. URL: <https://doi.org/10.1109/jphot.2020.3010966> (date of access: 09.07.2026).

25. Lee J., Hwang K.-i. YOLO with adaptive frame control for real-time object detection applications. *Multimedia tools and applications*. 2021. URL: <https://doi.org/10.1007/s11042-021-11480-0> (date of access: 09.07.2026).

26. An experiment-based review of low-light image enhancement methods / W. Wang та ін. *IEEE access*. 2020. Vol. 8. P. 87884–87917. URL: <https://doi.org/10.1109/access.2020.2992749> (date of access: 04.07.2026).

27. Li H., Xu K. Innovative adaptive edge detection for noisy images using wavelet and Gaussian method. *Scientific reports*. 2025. Vol. 15, № 1. URL: <https://doi.org/10.1038/s41598-025-86860-9> (date of access: 04.07.2026).

28. Blachut K., Kryjak T. Real-Time efficient FPGA implementation of the multi-scale lucas-kanade and horn-schunck optical flow algorithms for a 4K video stream. *Sensors*. 2022. Vol. 22, № 13. P. 5017. URL: <https://doi.org/10.3390/s22135017> (date of access: 04.07.2026).

29. The Segment Anything Model (SAM) for remote sensing applications: from zero to one shot / L. P. Osco та ін. *International journal of applied earth observation and geoinformation*. 2023. Vol. 124. P. 103540. URL: <https://doi.org/10.1016/j.jag.2023.103540> (date of access: 04.07.2026).

30. Jakhar Y., Borah M. D. Effective near-duplicate image detection using perceptual hashing and deep learning. *Information processing & management*. 2025. Vol. 62, № 4. P. 104086. URL: <https://doi.org/10.1016/j.ipm.2025.104086> (date of access: 04.07.2026).

31. Fast and accurate image super-resolution with deep laplacian pyramid networks / W.-S. Lai та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2019. Vol. 41, № 11. P. 2599–2613. URL: <https://doi.org/10.1109/tpami.2018.2865304> (date of access: 04.07.2026).

32. Zou B., Qiu H., Lu Y. Point cloud reduction and denoising based on optimized downsampling and bilateral filtering. *IEEE access*. 2020. Vol. 8. P. 136316–136326. URL: <https://doi.org/10.1109/access.2020.3011989> (date of access: 04.07.2026).

33. Hermens F. Automatic object detection for behavioural research using YOLOv8. *Behavior research methods*. 2024. URL: <https://doi.org/10.3758/s13428-024-02420-5> (date of access: 04.07.2026).
34. Sparse label assignment for oriented object detection in aerial images / Q. Ming та ін. *Remote sensing*. 2021. Vol. 13, № 14. P. 2664. URL: <https://doi.org/10.3390/rs13142664> (date of access: 04.07.2026).
35. Activated gradients for deep neural networks / M. Liu та ін. *IEEE transactions on neural networks and learning systems*. 2021. P. 1–13. URL: <https://doi.org/10.1109/tnnls.2021.3106044> (date of access: 04.07.2026).
36. Extended feature pyramid network for small object detection / C. Deng та ін. *IEEE transactions on multimedia*. 2021. P. 1. URL: <https://doi.org/10.1109/tmm.2021.3074273> (date of access: 04.07.2026).
37. Adaptive unscented kalman filter for target tracking with unknown time-varying noise covariance / B. Ge та ін. *Sensors*. 2019. Vol. 19, № 6. P. 1371. URL: <https://doi.org/10.3390/s19061371> (date of access: 04.07.2026).
38. Xiao X., Feng X. Multi-Object pedestrian tracking using improved yolov8 and OC-SORT. *Sensors*. 2023. Vol. 23, № 20. P. 8439. URL: <https://doi.org/10.3390/s23208439> (date of access: 04.07.2026).
39. Khan Z., Yoon S.-C., Bhandarkar S. M. Deep learning model compression and hardware acceleration for high-performance foreign material detection on poultry meat using NIR hyperspectral imaging. *Sensors*. 2025. Vol. 25, № 3. P. 970. URL: <https://doi.org/10.3390/s25030970> (date of access: 04.07.2026).
40. Adaptive two-tier deep learning for content-based image retrieval and classification with dynamic similarity fusion / A. M. Humadi та ін. *IET image processing*. 2025. Vol. 19, № 1. URL: <https://doi.org/10.1049/ipr2.70192> (date of access: 04.07.2026).
41. Kashyap R. The perfect marriage and much more: combining dimension reduction, distance measures and covariance. *Physica A: statistical*

mechanics and its applications. 2019. Vol. 536. P. 120938. URL: <https://doi.org/10.1016/j.physa.2019.04.174> (date of access: 04.07.2026).

42. Chen H., Shi Z. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote sensing*. 2020. Vol. 12, № 10. P. 1662. URL: <https://doi.org/10.3390/rs12101662> (date of access: 04.07.2026).

43. Rainio O., Teuho J., Klén R. Evaluation metrics and statistical tests for machine learning. *Scientific reports*. 2024. Vol. 14, № 1. URL: <https://doi.org/10.1038/s41598-024-56706-x> (date of access: 04.07.2026).

44. Instant neural graphics primitives with a multiresolution hash encoding / T. Müller та ін. *ACM transactions on graphics*. 2022. Vol. 41, № 4. P. 1–15. URL: <https://doi.org/10.1145/3528223.3530127> (date of access: 04.07.2026).

45. Sumaiya Thaseen I., Aswani Kumar C. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of king saud university – computer and information sciences*. 2017. Vol. 29, № 4. P. 462–472. URL: <https://doi.org/10.1016/j.jksuci.2015.12.004> (date of access: 04.07.2026).

46. Cai D., Zhang Z., Zhang Z. Corner-Point and foreground-area iou loss: better localization of small objects in bounding box regression. *Sensors*. 2023. Vol. 23, № 10. P. 4961. URL: <https://doi.org/10.3390/s23104961> (date of access: 04.07.2026).

47. Stergiou A., Poppe R. AdaPool: exponential adaptive pooling for information-retaining downsampling. *IEEE transactions on image processing*. 2022. P. 1. URL: <https://doi.org/10.1109/tip.2022.3227503> (date of access: 04.07.2026).

48. Figueiredo R. B. D., Mendes H. A. Analyzing information leakage on video object detection datasets by splitting images into clusters with high spatiotemporal correlation. *IEEE access*. 2024. P. 1. URL: <https://doi.org/10.1109/access.2024.3383047> (date of access: 04.07.2026).

49. Kim M., Finn C., Liang P. Fine-tuning vision-language-action models: optimizing speed and success. *Robotics: science and systems*. 2025. URL: <https://doi.org/10.15607/rss.2025.xxi.017> (date of access: 04.07.2026).

50. Pruning and quantization for deep neural network acceleration: a survey / T. Liang та ін. *Neurocomputing*. 2021. Vol. 461. P. 370–403. URL: <https://doi.org/10.1016/j.neucom.2021.07.045> (date of access: 04.07.2026).