

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

Автоматизована система підтримки прийняття рішень для супроводу та ідентифікації об'єктів у просторі на основі комп'ютерного зору та аналітики подій

Виконав: Мороз Дмитро Валентинович, студент 2-го року навчання ОП «Комп'ютерні науки»

Науковий керівник: Волинець Є. А., старший викладач

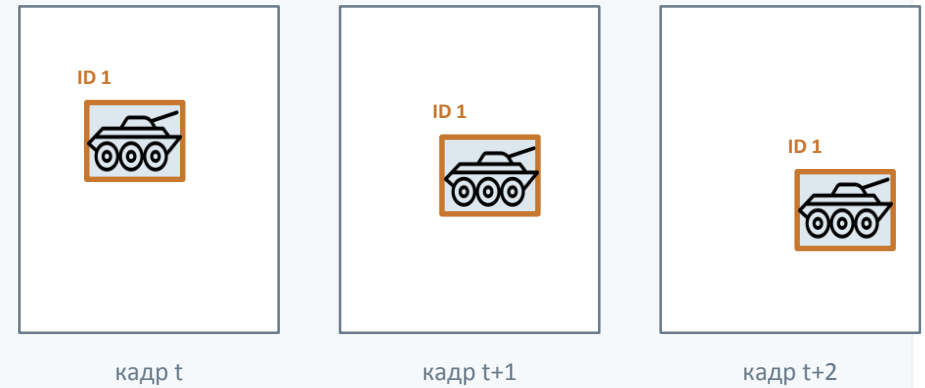
Проблематика: втрата зв'язку в умовах дії РЕБ-систем

- Засоби радіоелектронної боротьби глушать канали керування та навігаційні сигнали (GPS/GNSS), а також здійснюють спуфінг координат.
- Окрема проблема — радіогоризонт: на малих висотах радіохвилі не поширюються за межі прямої видимості (будівлі, дерева, ландшафт тощо можуть створювати перешкоди керуванню).
- Критичний момент для ударних БПЛА — кінцеве зниження на ціль: апарат опускається нижче й втрачає зв'язок саме на найвідповідальнішому етапі.
- **Втрата зв'язку перетворює дрон на некерований об'єкт → потрібна автономна бортова обробка інформації.**

Постановка задачі

- Розробити автономну бортову систему ідентифікації та відстеження наземних об'єктів, орієнтовану на роботу без зовнішнього зв'язку.
- Стабільна локалізація цілей у 2D-просторі пікселів — параметри обмежувальної рамки (Bounding Box).
- Безперервний трекінг з присвоєнням унікального ID для збереження цілісності спостереження між кадрами.
- **Результат → формування керуючого сигналу на центр обраної цілі: точне наведення та автоматизація оперативної обізнаності.**

BBox + стійкий ID між кадрами



Один і той самий ID супроводжує ціль попри рух камери та зміну ракурсу

Загальна архітектура: tracking-by-detection

Система складається з двох модулів: детектор знаходить об'єкти на кожному кадрі, трекер зіставляє детекції з існуючими треками та підтримує ID.



Детектор — YOLO11n (nano)

- Найлегша версія сімейства YOLO11 — мінімум обчислень
- Навчена на власному датасеті, єдиний клас «ціль»
- Придатна для портування на обладнання з обмеженими ресурсами

Трекер — KPByteTrack (власна модифікація)

- База — ByteTrack з бібліотеки BoxMOT
- Розширення: підтримка треків ключовими точками (AKAZE / ORB + RANSAC)
- Конфігурація через YAML-файл

Збір та диверсифікація даних

855

відеофрагментів

88.1 хв

сумарний хронометраж

~90%

реальні бойові дії

- Джерело — відкриті матеріали (Reddit), пошук за тематичними спільнотами для балансу класів.
- Свідома диверсифікація: усі пори року та погода (сніг, дощ, туман); три типи місцевості (відкрита, сільська, міська ~5–7%).
- Дві платформи зйомки: ударні FPV-дрони (динамічна камера) та розвідники (стабілізована) → переважно короткі треки.
- Специфіка: дим, пошкоджена/палаюча техніка, розмитість через рух.



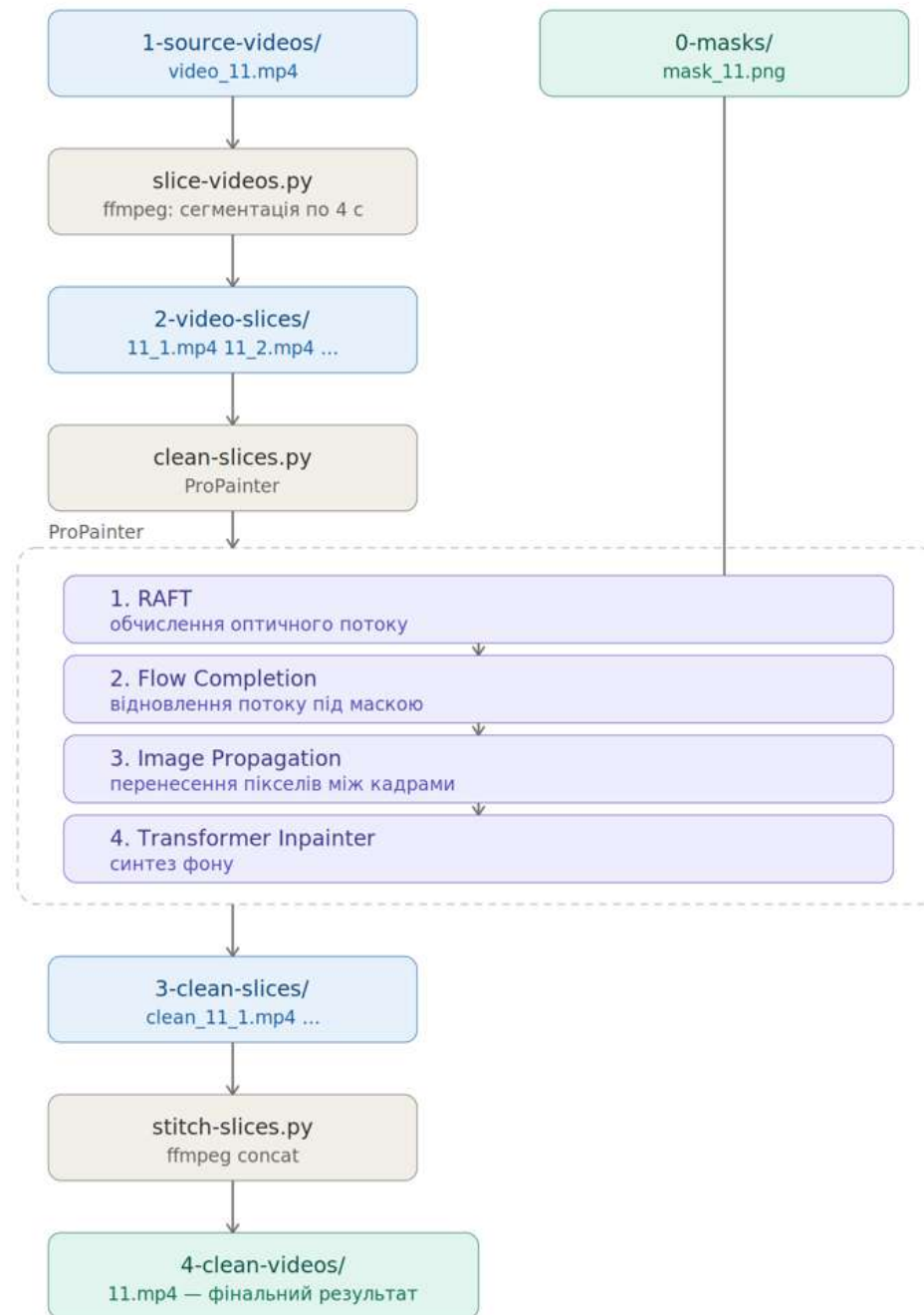
Конвеєр очищення відео

Навіщо: Detector Bias

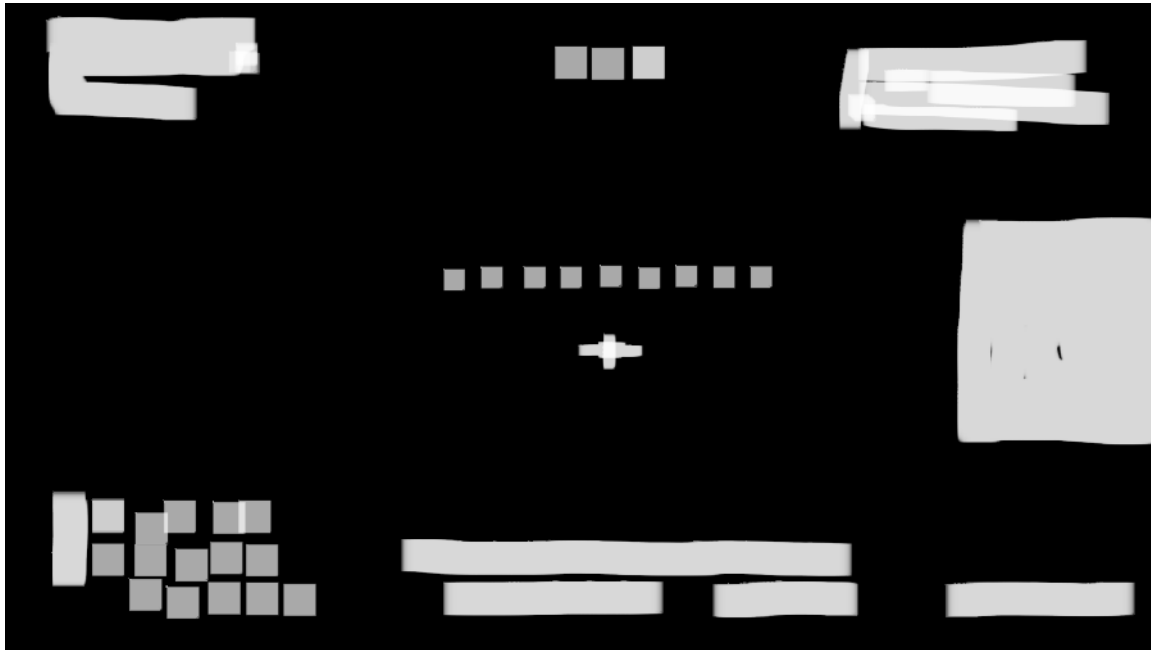
Інтерфейс дрона (OSD), приціли та логотипи статично лежать у центрі кадру — там, де зазвичай ціль. Мережа починає сприймати їх як ознаку об'єкта → хибні спрацювання (false positives).

- Вхід: відеофайл + бінарна PNG-маска (біле — зона видалення).
- slice-videos.py → сегменти по 4 с (обмеження VRAM: ProPainter тримає кадри в пам'яті GPU).
- clean-slices.py → інпейнтинг ProPainter (RAFT → Flow Completion → перенесення пікселів → трансформер).
- stitch-slices.py → склейка ffmpeg concat без перекодування.

*** Зрештою, обробка 88 хв відео зайняла приблизно 20 годин безперервної роботи комп'ютера.**



Приклад маски для видалення елементів з відео



Результат: до та після очищення

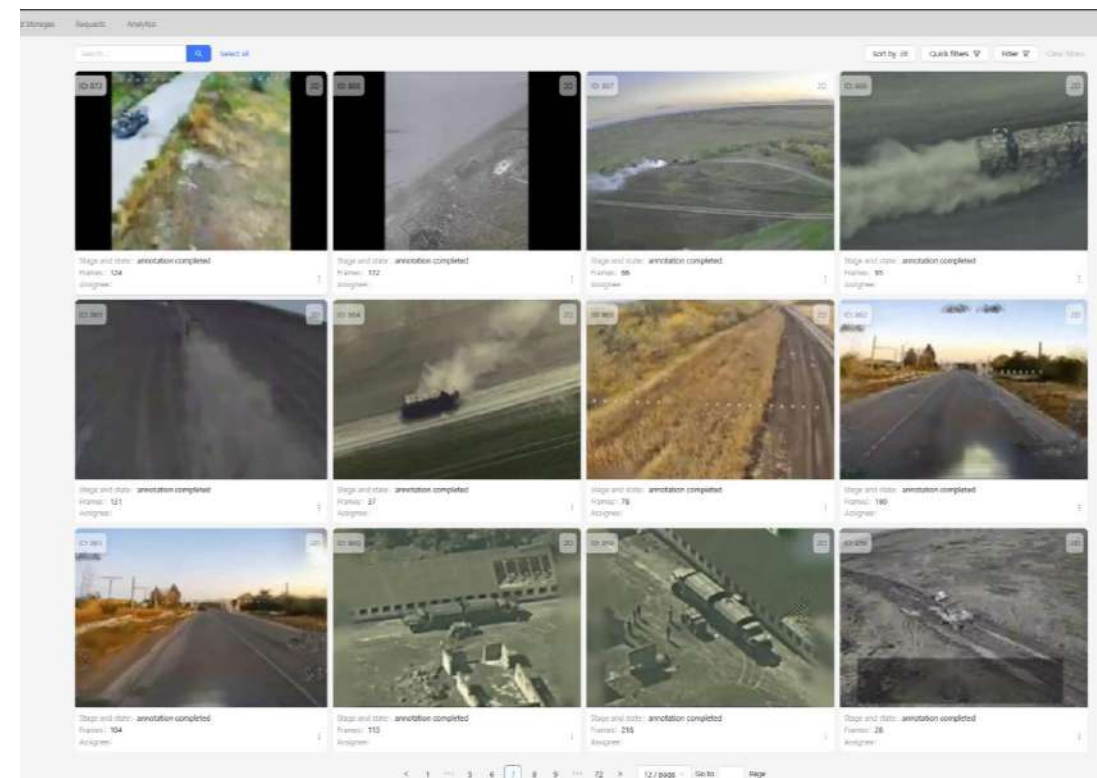


Зліва — сирий кадр з накладеним інтерфейсом та логотипами. Справа — той самий кадр після нейромережевого інпейнтингу: статичні артефакти видалено, фон контекстуально відновлено.

Анотація відеоматеріалів у CVAT

- Локальне розгортання CVAT у Docker — безкоштовний, з відкритим кодом, паралельна робота кількох користувачів.
- Розмітка на рівні відеопослідовностей: кожному об'єкту — клас і унікальний ID треку в межах відео.
- Ключова функція — інтерполяція рамок між кадрами: рамка ставиться на початку й кінці треку, проміжні CVAT добудовує сам.
- Експорт у форматі CVAT for video 2.1 (.xml) → власний Python-скрипт конвертує в індивідуальні .csv.

Трек — послідовність анотованих рамок одного й того самого об'єкта впродовж послідовних кадрів, з унікальним ID у межах відео.



«Розумна» екстракція кадрів з відео

Сусідні кадри відео майже ідентичні → пряме навчання веде до перенавчання. Власний скрипт відбирає інформативні кадри у 4 фази:

1 · Event-driven

Безумовно: перша/остання поява об'єкта, зміна occluded, перехід між scale-бакетами.

2 · Scale-дедуплікація

Бакети tiny/small/medium/large окремо; відбір за порогом IoU між сусідніми кадрами.

3 · Глобальна дедуплікація

Усунення майже-дублікатів, що потрапили через кілька бакетів (за IoU треку).

4 · Перцептивний хеш (pHash)

Фільтр візуальної схожості за відстанню Хеммінга; event-кадри обходять його.

7 115

відібрано кадрів (з 117 108) — 6.02%

6 055

train

752 / 308

val / test

262

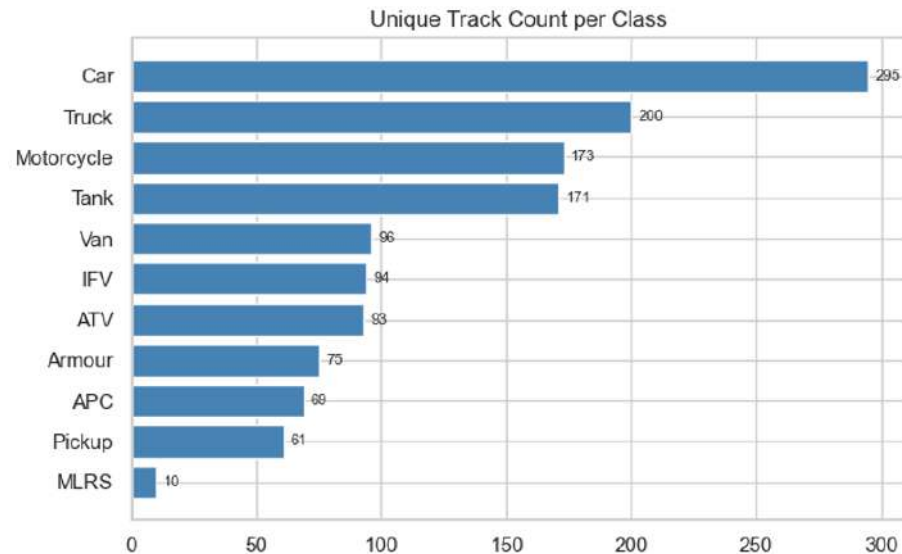
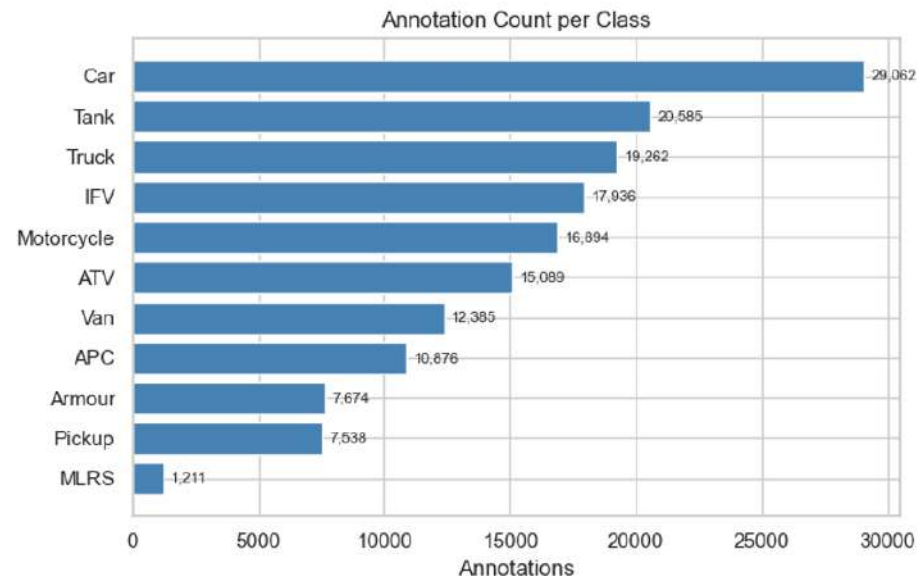
фонові кадри без об'єктів

Розбивка train / val / test — вручну: різні відео можуть містити ті самі локації, тому випадковий поділ спотворив би оцінку.

Аналіз відео-датасету та рішення про класи

- 1 337 унікальних треків, 158 512 анованих кадрів; медіанна тривалість треку — 75 кадрів (короткі через FPV-зйомку).
- Сильне переважання дрібних об'єктів: 81.6% анотацій — категорія Tiny (до 50x50 px).
- 11 класів CVAT згруповано у 4 супер-класи (Armor, Truck, Car, LightVehicle) через візуальну схожість і дисбаланс (MLRS — лише 10 треків).

Фінальне рішення → єдиний клас «ціль». За обмеженого обсягу даних пріоритет — надійна локалізація, а не класифікація.

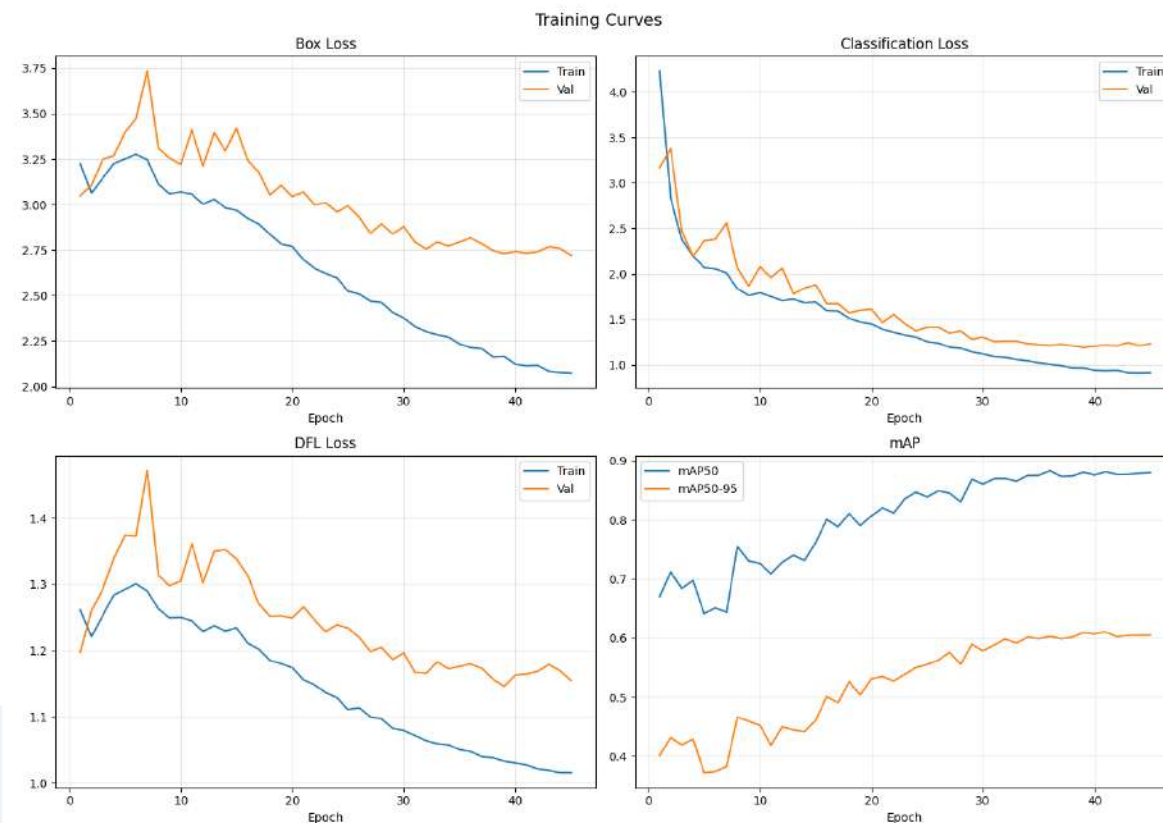


Підбір гіперпараметрів та навчання

- Optuna з TPE-семплером (байєсівська оптимізація): 30 випробувань × 40 epoch, patience = 7.
- Цільова метрика — mAP50-95 на валідації;
- `imgsz = 800 px` замість 640 — щоб зберегти деталі дрібних об'єктів (81.6% датасету).
- Аугментації критичні через малий обсяг і кореляцію кадрів у межах відео: HSV, геометрія, mosaic, mixup, copy-paste.

$$L = \text{box} \cdot L_{\text{CIoU}} + \text{cls} \cdot L_{\text{BCE}} + \text{dfl} \cdot L_{\text{DFL}}$$

CIoU враховує відстань центрів і пропорції; DFL — anchor-free регресія рамки через розподіл.



Показники детектора (єдиний клас «ціль»)

0.881

mAP@50

0.611

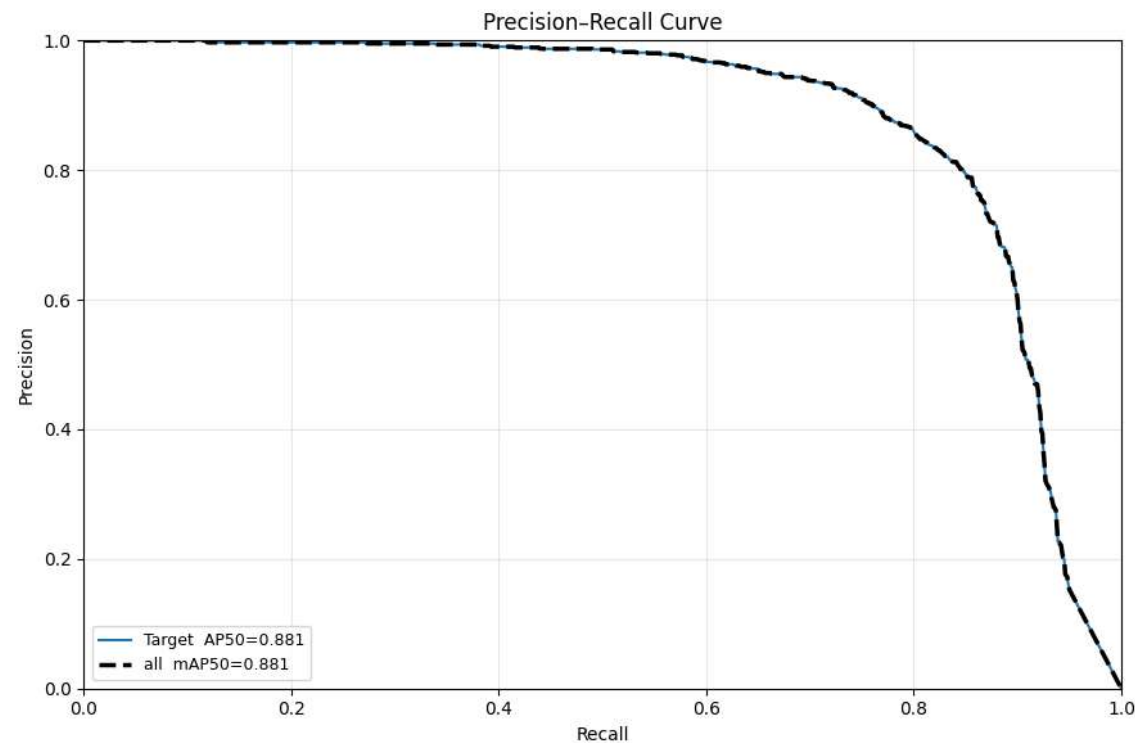
mAP@50-95

0.856

Precision

Детекція малих об'єктів (area < 2500 px², IoU ≥ 0.5)

- Recall = 0.647 — виявлено 123 з 190 малих цілей. Чим раніше виявлено ціль, тим більше часу на рішення.
- Precision нижча (0.369) — типово для дрібних об'єктів: текстура фону провокує хибні спрацювання.
- Для задачі наведення Recall пріоритетніший: пропустити ціль гірше, ніж отримати зайву детекцію, яку трекер відфільтрує.



1 клас vs 4 класи: class-agnostic порівняння

Питання: чи 4-класовий детектор гірше ЛОКАЛІЗУЄ, чи лише гірше КЛАСИФІКУЄ? Оцінили обидва без урахування класу — лише за здатністю знайти об'єкт.

Метрика (class-agnostic)	1 клас	4 класи
True Positives (TP)	1034	1032
False Negatives (FN)	224	226
False Positives (FP)	194	169
Precision	0.842	0.859
AP50	0.863	0.859

Висновок

- Обидва детектори локалізують практично рівноцінно: AP50 0.859 vs 0.863, FN майже однакові.
- Проблема 4-класового варіанту — виключно класифікація, а не виявлення об'єктів.
- ByteTrack не використовує клас при асоціації — йому потрібні лише BBox. Тому для трекінгу обидва однаково придатні.

Алгоритм ByteTrack

- Парадигма tracking-by-detection без ознак зовнішнього вигляду — лише перетин BBox.
- Фільтр Калмана прогнозує наступну позицію треку; вартість зіставлення — IoU-відстань; призначення — угорський алгоритм.
- Головна ідея — двоетапна асоціація замість відкидання слабких детекцій:

Етап 1: детекції з високою впевненістю → активні треки (IoU).

Етап 2: низьковпевнені детекції → треки без пари з етапу 1. Це відновлює об'єкти, коли детектор тимчасово «невпевнений».



(a) detection boxes



(b) tracklets by associating high score detection boxes



(c) tracklets by associating every detection box

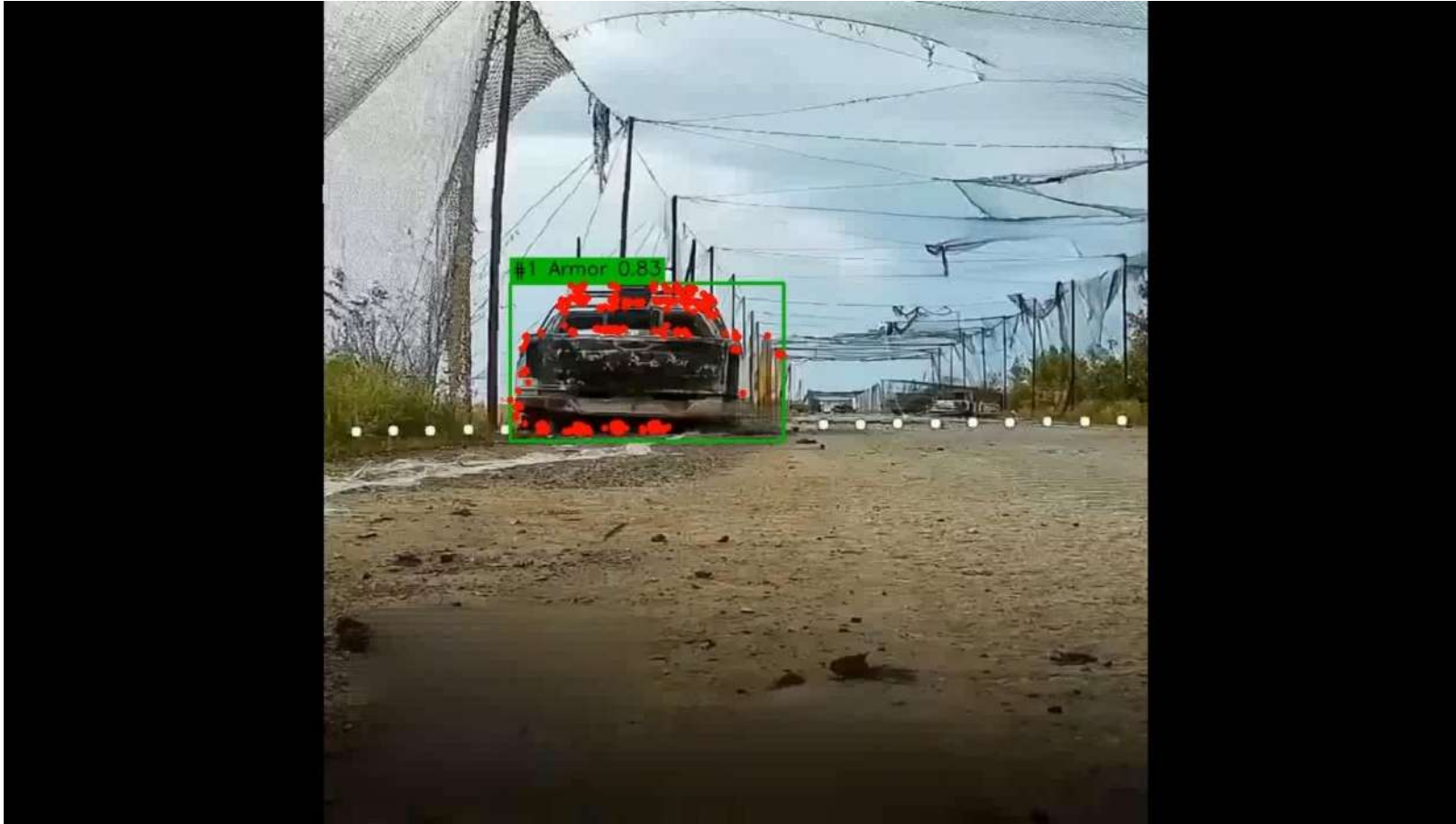
КРByteTrack — підтримка треків ключовими точками

Розширення поверх ByteTrack (BoxMOT): компенсує пропуски детектора, синтезуючи рамку з геометрії ключових точок.



Ефект: навіть коли детектор пропускає об'єкт, трек «дотягується» синтезованою рамкою — підтримка ідентифікації не переривається.
Гіпотеза: користь зростає у випадках коли детектор не продукує детекції через низьку натренованість або занадто високий встановлення поріг впевненості.

Візуалізація роботи алгоритму трекінгу



Метрики багатооб'єктного відстеження

Метрика	Що оцінює
IDF1	Стійкість ідентифікації об'єкта — першочергова метрика
HOTA	Баланс детекції та асоціації = $v(\text{DetA} \cdot \text{AssA})$
DetA / AssA	Якість виявлення / якість зіставлення треків окремо
MOTA / MOTP	Агрегована якість / просторова точність рамок
FP / FN / IDSW	Хибнопозитивні / пропущені / переключення ID

Агрегація: сумуємо сирі лічильники (TP, FP, FN...) по всіх відео, потім перераховуємо метрики. Довші/складніші відео отримують справедливу вагу.

Інструмент: TrackEval (протоколи MOT Challenge). Метрики рахуються окремо для кожного з тестових відео, потім агрегуються.

video	MOTA	MOTP	IDF1	HOTA	DetA	AssA	IDSW	FP	FN	MT	ML
2	1.000000	0.894176	1.000000	0.897410	0.897410	0.897410	0	0	0	1	0
4	1.000000	0.844307	1.000000	0.846131	0.846131	0.846131	0	0	0	1	0
54	0.760870	0.852363	0.791209	0.650392	0.705944	0.599357	4	8	10	1	0
74	-0.564706	0.812902	0.247826	0.235721	0.233284	0.239679	2	192	72	0	0
79	0.528302	0.792909	0.691358	0.536755	0.418997	0.687775	0	0	50	1	1
81	0.769231	0.823863	0.782178	0.655427	0.649896	0.664289	1	20	0	1	0
93	0.974066	0.868834	0.987179	0.847401	0.840778	0.854231	0	32	2	8	0
100	0.538336	0.812063	0.240137	0.283311	0.553646	0.147227	27	98	158	3	0
101	1.000000	0.825496	1.000000	0.843389	0.843389	0.843389	0	0	0	1	0
117	-0.250000	0.913697	0.409938	0.323203	0.282112	0.381605	0	52	43	0	0
118	0.767442	0.766420	0.895833	0.673851	0.608861	0.745833	0	10	0	1	0
119	0.937500	0.814928	0.967742	0.743920	0.743920	0.743920	0	0	4	1	0
125	0.747525	0.796664	0.848485	0.603018	0.599794	0.606666	2	4	45	1	0
142	0.802632	0.805593	0.525000	0.494003	0.664424	0.367851	1	11	3	1	0
157	1.000000	0.851215	1.000000	0.830125	0.830125	0.830125	0	0	0	1	0
170	0.976190	0.831173	0.987952	0.813541	0.813541	0.813541	0	0	3	1	0

ByteTrack vs KPByteTrack: перевірка гіпотези

Детектор ключових точок — ORB.

Експеримент 1 — якісний детектор			Експеримент 2 — недотренований детектор		
<i>mAP50 = 0.881</i>			<i>mAP50 = 0.762</i>		
IDF1		+0.0015	IDF1		+0.0175 (×12)
IDSW		108 → 82	HOTA		+0.0116
FN		4178 → 4066	FN		6878 → 6287
FP		1797 → 2435	FP		1956 → 2621

Швидкодія — головне обмеження

Показник	ByteTrack	KPByteTrack (ORB)
Пропускна здатність	92.6 FPS	12.1 FPS
Коефіцієнт реального часу	3.09×	0.40×
Час обробки (13 149 кадрів)	2 хв 21 с	18 хв 02 с

Обмеження дослідження

- Один датасет
- Один дескриптор (ORB)
- Тільки два рівні якості детектора

Компроміс: вища якість відстеження ціною реального часу. Падіння $\sim 8\times$ — через екстракцію/зіставлення точок і RANSAC на кожному треку щокадру.

→ Результати коректно читати як стійку тенденцію, що підтверджує гіпотезу, а не як доведену універсальну закономірність.

Висновки

- Спроектовано й реалізовано автономну систему ідентифікації та відстеження наземних об'єктів на основі комп'ютерного зору.
- Сформовано власний датасет (855 відео), розроблено конвеєр очищення (ProPainter) та 4-фазний алгоритм екстракції кадрів.
- Навчено детектор YOLO11n (mAP50 = 0.881); доведено, що різниця 1 vs 4 класи — лише в класифікації, не в локалізації.
- Розроблено KPByteTrack; експерименти підтвердили гіпотезу: користь модифікації зростає зі зниженням якості детектора (IDF1 +0.0015 → +0.0175).
- Виявлено обмеження — швидкодію.

Перспективи розвитку

Швидкодія

Квантування + експорт TensorRT (NVIDIA Jetson); переписання критичних компонентів на C++; розпаралелювання обробки точок.

Детектор

Розширення та балансування датасету (MLRS, APC, урбаністика); метрики стабільно зростали — плато не досягнуто.

Алгоритм трекінгу

Систематичне порівняння AKAZE vs ORB; кілька рівнів якості детектора; повторні прогони для статистики.

Прикладні напрями

Інтеграція з бортовим обладнанням і польові випробування; оцінка відстані до цілі; повернення до класифікації.

Дякую за увагу! Готовий відповісти на запитання.