

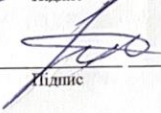


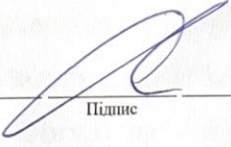
Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень

Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент групи КН-22-2  Олексій КУЛІНІЧ  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ  
Керівник: к.т.н., доц. каф. КН  Олександр ПАСІЧНИК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ  
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
Зав. кафедри КН, д.т.н., професор  Олександр БАРМАК  
Підпис Ім'я, ПРІЗВИЩЕ

10 6 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь бакалавр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

  
(підпис)

д.т.н., професор Олександр БАРМАК  
« 22 » січня 2026 року

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень»
2. Завдання видано студенту Олексію Кулінічу  
(Ім'я, прізвище)
3. Керівник роботи доцент кафедри КН Олександр Пасічник  
(посада, ім'я, прізвище)
4. Затверджено наказом університету від « 20 » січня 2026 р. № 7
5. Дата видачі завдання студенту: « 22 » січня 2026 р.
6. Зміст тексту кваліфікаційної роботи (перелік задач) та вихідні дані:

Мета кваліфікаційної роботи – підвищення точності ідентифікації та швидкості виявлення наземних роботизованих комплексів на зображеннях шляхом розроблення нейромережевого методу. Завдання кваліфікаційної роботи: провести аналіз сучасного стану технологій та підходів до ідентифікації наземних роботизованих комплексів за допомогою методів комп'ютерного зору та глибокого навчання; формалізувати задачу ідентифікації наземних роботизованих комплексів як задачу детекції об'єктів на зображенні; розробити метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень, спрямований на підвищення точності та швидкості виявлення; виконати програмну реалізацію методу та провести експериментальне дослідження.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання	січень 2026	виконано
2	Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження	лютий 2026	виконано
3	Проектування методу розв'язання задачі, опис архітектурних рішень, розроблення математичних моделей та алгоритмів	березень 2026	виконано
4	Обґрунтування інструментарію розробки, програмна реалізація розробленого методу, проведення експериментального тестування та оцінювання ефективності	квітень 2026	виконано
5	Написання тексту кваліфікаційної роботи, урахування зауважень керівника, оформлення згідно з вимогами	травень 2026	виконано
6	Розроблення презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2026	виконано
7	Отримання відгуку керівника, рецензії, перевірка тексту кваліфікаційної роботи на плагіат, нормоконтроль	червень 2026	виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2026	виконано

Виконавець: студент групи КН-22-2  
Група виконавця

  
Підпис

Олексій КУЛІНІЧ  
Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  
Науковий ступінь, посада

  
Підпис

Олександр ПАСТІЧНИК  
Ім'я, ПРІЗВИЩЕ

### Анотація

Тема кваліфікаційної роботи бакалавра: «Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-22-2  
Олексій Кулініч


Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН  
Олександр Пасічник

Кваліфікаційна робота бакалавра містить:

Текст кваліфікаційної роботи				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
68	16	4	46	2

Мета кваліфікаційної роботи – підвищення точності ідентифікації та швидкості виявлення наземних роботизованих комплексів на зображеннях шляхом розроблення нейромережевого методу. Розроблено нейромережевий метод ідентифікації, що включає математичну формалізацію задачі, вибір архітектури нейронної мережі, опис конвеєру обробки даних. Виконано програмну реалізацію запропонованого методу у вигляді прототипного застосунку мовою програмування Python з використанням сучасних фреймворків глибокого навчання. Проведено експериментальну оцінку ефективності методу на репрезентативних тестових наборах даних з розрахунком метрик якості (точність, recall, F1-score, mAP), із використанням архітектур RT-DETR та YOLO.

Ключові слова: детекція об'єктів, глибоке навчання, комп'ютерний зір, нейромережеві архітектури, наземні роботизовані комплекси, RT-DETR, YOLO.

Виконавець: студент групи КН-22-2  Олексій КУЛІНІЧ  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	3
Вступ .....	5
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій ..8	
1.1 Аналіз інформаційних моделей ідентифікації НРК .....	8
1.2 Огляд теоретичних підходів до ідентифікації об'єктів.....	10
1.3 Аналіз існуючих наукових рішень .....	12
1.4 Аналіз існуючих програмних рішень.....	15
1.5 Мета та завдання роботи .....	17
Розділ 2 Проектування нейромережевого методу ідентифікації наземних роботизованих комплексів за аналізом зображень.....	18
2.1 Концепція та етапи нейромережевого методу ідентифікації.....	18
2.2 Алгоритмічна модель нейромережевого методу .....	19
2.3 Архітектура нейромережі.....	22
2.4 Алгоритм навчання нейронної мережі .....	26
2.5 Інформаційна модель вхідних даних .....	27
2.6 Характеристика датасету.....	28
2.7 Метрики оцінювання .....	29
2.8 Висновки до розділу 2.....	32
Розділ 3 Розробка та експериментальна перевірка методу нейромережевої ідентифікації НРК.....	33
3.1 Проектна архітектура програмної реалізації та функціональна взаємодія компонентів.....	33
3.1.1 Особливості використання спеціалізованих програмних компонентів.....	34
3.1.2 Компоненти програмного комплексу .....	35
3.2 Реалізація експериментального дослідження .....	36
3.2.1 Навчання нейронної мережі та перевірка на валідаційній вибірці.....	37
3.2.2 Тестування методу .....	42
3.3 Висновки до розділу 3.....	49
Загальні висновки .....	51
Перелік посилань .....	54
Додатки	

### Перелік скорочень

Скорочення, термін, позначення	Пояснення
RT-DETR	Real-Time Detection Transformer – трансформерна архітектура детекції в реальному часі.
КН	Комп’ютерні науки
API	Application Programming Interface – програмний інтерфейс застосунків
MLP	Multi-Layer Perceptron – багатошаровий перцептрон
НРК	Наземний роботизований комплекс
CCFM	Cross-scale Feature-fusion Module – модуль міжмасштабного злиття ознак
CNN	Convolutional Neural Network – згорткова нейронна мережа
GPU	Graphics Processing Unit – графічний процесор
CUDA	Compute Unified Device Architecture – програмно-апаратна платформа NVIDIA для GPU-обчислень
AMP	Automatic Mixed Precision – автоматичне навчання зі змішаною точністю
FPS	Frames Per Second – кадрів за секунду
mAP	Mean Average Precision – середня точність детекції
IoU	Intersection over Union – відношення площі перетину до площі об’єднання рамок
NMS	Non-Maximum Suppression – алгоритм придушення немаксимальних спрацювань
RGB	Red, Green, Blue – колірна модель з трьома каналами

AIFI	Intra-scale Feature Interaction – внутрішньомасштабна взаємодія ознак
HOG	Histogram of Oriented Gradients – гістограма орієнтованих градієнтів
YOLO	You Only Look Once – сімейство одноетапних детекторів об'єктів
DETR	Detection Transformer – трансформерна архітектура для детекції об'єктів
ViT	Vision Transformer – візуальний трансформер
FPN	Feature Pyramid Network – мережа піраміди ознак
TP	True Positive – істинно позитивні спрацювання
FP	False Positive – хибнопозитивні спрацювання
FN	False Negative – хибнонегативні пропуски
CSV	Comma-Separated Values – текстовий формат подання даних
SIFT	Scale-Invariant Feature Transform – масштабоінваріантне ознакове перетворення

## Вступ

Кваліфікаційна робота присвячена розробці нейромережевого методу ідентифікації наземних роботизованих комплексів за аналізом зображень.

**Актуальність теми.** Розвиток методів глибокого навчання та комп'ютерного зору суттєво розширив можливості автоматизації задач, які раніше вважалися такими, що потребують безпосереднього контролю людини-оператора. Особливо відчутним цей зсув є у сферах, де ціна затримки або помилки надто висока: спостереження за полем бою, навігація автономних систем, ситуаційна обізнаність у реальному часі. Одним із таких прикладних напрямків є автоматизована ідентифікація наземних роботизованих комплексів (НРК) – завдання, що набуває стратегічного значення в умовах широкого впровадження автономних систем у різні сфери. Ефективна автоматизована ідентифікація таких об'єктів на зображеннях є критичною для забезпечення ситуаційної обізнаності, безпеки навігації та координації дій. Основні виклики включають: варіативність ракурсів, камуфляж, подібність форм до цивільної техніки, динамічне освітлення, погодні умови та перешкоди (дим, пил, рослинність). Традиційні методи комп'ютерного зору (наприклад, на основі HOG, SIFT) мають низьку точність у складних сценах, тоді як сучасні архітектури глибокого навчання (YOLO, DETR, RT-DETR тощо) дозволяють досягти високої швидкодії та точності, але потребують адаптації під специфіку НРК. Розробка спеціалізованого нейромережевого методу, що поєднує високу точність детекції з мінімальною затримкою обробки, є актуальним завданням комп'ютерних наук, яке сприяє підвищенню ефективності автономних систем у реальних умовах. В Україні тематика наземних роботизованих комплексів активно розвивається через державний кластер оборонних технологій Brave1 [1].

Станом на поточний момент в російсько-українській війні ворогом використовується щонайменше 20 типів НРК у бойових умовах, а аналітики ідентифікують 32 моделі російських НРК. Водночас, Україна кодифікувала для Сил оборони понад 70 наземних роботизованих платформ і фінансує вже близько

200 найменувань. За типом призначення використовується понад 10 основних класів НРК, що перетворює характер ведення бойових дій у роботизовану площину, а успішне протистояння визначається точним та швидким виявленням ворожих НРК з подальшим прийняттям оперативних рішень.

**Об’єкт дослідження** – процес ідентифікації наземних роботизованих комплексів на основі аналізу зображень.

**Предмет дослідження** – методи та алгоритми детекції об’єктів на зображенні.

**Мета кваліфікаційної роботи** – підвищення точності ідентифікації та швидкості виявлення наземних роботизованих комплексів на зображеннях шляхом розроблення нейромережевого методу.

**Завдання кваліфікаційної роботи:**

– провести аналіз сучасного стану технологій та підходів до ідентифікації наземних роботизованих комплексів за допомогою методів комп’ютерного зору та глибокого навчання;

– формалізувати задачу ідентифікації наземних роботизованих комплексів як задачу детекції об’єктів на зображенні;

– розробити метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень, спрямований на підвищення точності та швидкості виявлення;

– виконати програмну реалізацію методу;

– провести експериментальне дослідження методу з використанням метрик точності та оцінкою швидкості виявлення.

Практичне значення одержаних результатів полягає у створенні програмного засобу, що реалізує нейромережевий метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень і забезпечує прийняття рішення без ручного аналізу.

Запропонований метод дозволяє Силам Оборони України оперативно отримувати об’єктивні дані, щодо наявності наземних роботизованих комплексів на ділянках відповідальності. Використання методів глибокого навчання,

комп'ютерного зору та алгоритмів детекції система автоматично забезпечує швидкий аналіз зображення та мінімізує вплив людського чинника. Автоматизація цього процесу пришвидшує отримання достовірної інформації щодо наявності або відсутності НРК та забезпечує підвищення стійкості Сил Оборони України при вирішенні завдань наземного моніторингу.

Практичною перевагою розробленої системи є її універсальність та доступність, яка ґрунтується на використанні зображень інваріантних щодо джерел походження у поєднанні з невибагливим до апаратного забезпечення програмним рішенням. У результаті засоби для об'єктивного прийняття рішення щодо поточної обстановки стає доступним для широкого кола фахівців різного рівня підпорядкування.

## **Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій**

### **1.1 Аналіз інформаційних моделей ідентифікації НРК**

Автоматизований аналіз візуальної інформації є важливим напрямом розвитку інтелектуальних інформаційних систем, що поєднує методи обробки даних, машинного навчання та комп'ютерного зору. Такі підходи дають змогу не лише виявляти й класифікувати об'єкти на зображеннях, а й підвищувати автономність технічних систем у складних умовах експлуатації. Розвиток технологій НРК та методів штучного інтелекту за останнє десятиліття зумовив значний прогрес у сфері автоматизованого виявлення та класифікації об'єктів у складних ландшафтних умовах. Інтеграція алгоритмів комп'ютерного зору та нейромережевих підходів дозволила досягти нового рівня автономності систем, що функціонують без безпосередньої участі оператора [2]. Використання методів глибокого навчання для ідентифікації НРК є критично важливим напрямком досліджень, зважаючи на необхідність забезпечення ситуаційної обізнаності, безпеки навігації та оперативного реагування в умовах динамічного середовища [3].

Наземний моніторинг із використанням мобільних сенсорних платформ набуває все більшого значення для широкого спектру застосувань, включаючи охорону територій, логістичні операції у важкодоступних районах та виконання завдань у зонах підвищеного ризику. Суміжним напрямом є детекція об'єктів із безпілотних літальних апаратів, де задача ідентифікації НРК набуває додаткової складності [4] через мінливий ракурс та малий кутівий розмір цілі [5]. Поєднання мобільності НРК з потужними можливостями алгоритмів комп'ютерного зору відкриває нові перспективи для створення інтелектуальних систем спостереження. На відміну від повітряного моніторингу, наземне спостереження характеризується вищою складністю через гетерогенність фону, наявність статичних та динамічних перешкод, а також різкі зміни умов освітлення, що суттєво ускладнює процес автоматизованої ідентифікації та

потребує розробки стійких інформаційних моделей.

Сучасні підходи до детекції та ідентифікації НРК охоплюють широкий спектр методів, які можуть бути структуровані за кількома ключовими напрямками: архітектури мереж для детекції об'єктів, методи попередньої обробки візуальних даних, специфіка виділення ознак у складних ландшафтах та підходи до навчання моделей в умовах обмежених навчальних вибірок. Важливим аспектом є подолання специфічних обмежень наземного середовища, таких як високий рівень зашумленості сигналу та низький контраст між цільовим об'єктом і фоном.

Інформаційна модель ідентифікації НРК включає три основні групи даних: вхідні статичні зображення, ознакове представлення об'єкта після обробки нейронною мережею та вихідний результат у вигляді класу, координат обмежувальної рамки і значення впевненості. У такій моделі початкове RGB-зображення є джерелом сирової інформації, а кінцевий вектор ідентифікації є стислим описом знайденого об'єкта. Для задачі НРК особливе значення має врахування контексту сцени. Один і той самий об'єкт може змінювати видимі контури залежно від ракурсу, освітлення, часткового перекриття або фону. Тому інформаційна модель повинна описувати не лише локальні ознаки корпусу, коліс, гусениць чи антен, а й просторові зв'язки між цими ознаками в межах усього кадру.

З точки зору задачі детекції НРК є об'єктом із характерним набором візуальних та геометричних ознак. Типовий малогабаритний НРК має компактний прямокутний або трапецієподібний корпус, гусеничний або колісний рушій (як правило, 4–6 коліс або дві гусениці), антенний модуль або сенсорну платформу зверху, а також характерні маркери чи кронштейни. Масштаб об'єкта у кадрі суттєво варіює залежно від відстані до камери: від кількох пікселів при зйомці з дрону до займання значної частини кадру при близькому огляді. Ключовою складністю є подібність зовнішнього вигляду НРК до цивільної або сільськогосподарської техніки: трактори, робочі платформи та промислові транспортні засоби мають схожий силует, що потребує від детектора не лише локальної класифікації форм, а й врахування контексту сцени та взаємного розташування ознак. Саме ця специфіка зумовлює перевагу

трансформерних архітектур з механізмом глобальної уваги над класичними CNN-детекторами при ідентифікації НРК.

Вихідна інформація має бути придатною для практичного використання оператором або іншою програмною підсистемою. Саме тому результат ідентифікації доцільно подавати не лише як координати рамки, а і як інтерпретований рівень достовірності. Це зменшує неоднозначність під час аналізу складних сцен і створює основу для подальшої програмної реалізації методу.

## **1.2 Огляд теоретичних підходів до ідентифікації об'єктів**

Системи ідентифікації об'єктів у комп'ютерному зорі базуються на методах машинного навчання, зокрема на парадигмі навчання з учителем, яка передбачає використання розмічених наборів даних [6]. У цьому підході модель навчається встановлювати відповідність між вхідними зображеннями та їх анотованими характеристиками, такими як клас об'єкта та його просторове розташування. Якість такої моделі напряму залежить від обсягу, різноманітності та точності розмітки навчального набору даних. Одним із ключових теоретичних аспектів є організація обробки вхідних даних, або Data Pipeline. Він включає етапи збору, очищення, нормалізації та аугментації зображень.

Аугментація є критично важливою, оскільки дозволяє штучно розширити навчальну вибірку за рахунок варіацій вхідних даних, таких як зміна освітлення, геометричні трансформації, додавання шуму або імітація погодних умов. Серед ефективних методів аугментації слід відзначити Copy-Paste [7]. Це знижує ризик перенавчання моделі та підвищує її здатність узагальнювати нові дані. Особливо це актуально для задач ідентифікації НРК, де об'єкти можуть частково перекриватися, змінювати масштаб або знаходитися у складному фоні. З точки зору архітектурних рішень, сучасні підходи до ідентифікації об'єктів можна поділити на кілька категорій: класичні згорткові нейронні мережі, гібридні моделі та трансформерні архітектури. Згорткові мережі добре працюють з локальними ознаками, однак їх ефективність знижується у випадках, коли для

розпізнавання необхідно враховувати глобальний контекст сцени. У зв'язку з цим зростає популярність трансформерних моделей, зокрема Vision Transformer (ViT) [8] та Swin Transformer [9], які використовують механізм уваги для аналізу взаємозв'язків між різними частинами зображення.

Особливістю трансформерних детекторів є відмова від класичних процедур постобробки, таких як немаксимальне придушення (NMS) [10]. Замість цього використовується механізм оптимального зіставлення передбачень моделі з еталонними об'єктами, що дозволяє зменшити кількість дублюючих детекцій та підвищити стабільність результатів. Це особливо важливо для систем реального часу, де затримка обробки повинна бути мінімальною. Окрему увагу слід приділити проблемі дисбалансу даних та наявності схожих за виглядом об'єктів [6]. У задачах ідентифікації НРК модель може помилково класифікувати цивільну або сільськогосподарську техніку як роботизований комплекс. Для зменшення кількості таких помилок застосовуються методи відбору складних негативних прикладів, які дозволяють моделі краще розрізняти подібні класи.

З практичної точки зору, реалізація подібних систем не потребує використання повноцінних реляційних баз даних, оскільки основний процес полягає у потоковій обробці зображень або відео. Використання класичних СКБД може призвести до додаткових затримок через накладні витрати на виконання запитів. Натомість більш ефективним є застосування легковагових механізмів збереження результатів, таких як логування у форматах JSON або CSV. Це дозволяє фіксувати ключові метрики роботи системи, зокрема точність ідентифікації, рівень впевненості моделі та час обробки кадру, без суттєвого впливу на продуктивність.

Програмна реалізація систем ідентифікації, як правило, будується за модульним принципом, що забезпечує гнучкість та можливість масштабування. Окремі компоненти відповідають за попередню обробку даних, інференцію моделі та аналіз результатів. Для обробки зображень можуть використовуватись спеціалізовані бібліотеки, які забезпечують ефективні операції над матрицями пікселів, включаючи зміну розміру, нормалізацію та фільтрацію шумів.

Окремим теоретичним напрямом є методи навчання з підкріпленням

(Reinforcement Learning) та активного навчання (Active Learning), які дозволяють мінімізувати обсяг необхідної розмітки шляхом інтелектуального відбору найбільш інформативних зразків для анотування [6]. У контексті ідентифікації НРК це є критично важливим, оскільки отримання великих розмічених вибірок спеціалізованих об'єктів є технічно складним та ресурсомістким процесом. Поряд із цим, синтетична генерація даних із застосуванням рушіїв фізичного моделювання (наприклад, Unreal Engine або Unity Perception) відкриває можливість масштабування навчальних вибірок без залучення реальних об'єктів, що є перспективним підходом для вирішення проблеми браку спеціалізованих даних.

Таким чином, теоретичні підходи до розв'язку задач ідентифікації об'єктів поєднують у собі методи машинного навчання, ефективну організацію даних та оптимізовані архітектурні рішення [6]. Їх правильне комбінування дозволяє створювати системи, здатні працювати в умовах реального часу та забезпечувати високу точність розпізнавання навіть у складних середовищах.

### **1.3 Аналіз існуючих наукових рішень**

Вибір архітектури нейромережі є ключовим рішенням при проєктуванні системи ідентифікації НРК, оскільки він визначає компроміс між точністю детекції, швидкістю та стійкістю до часткового перекриття об'єктів у складних умовах (камуфляж, зміна освітленості, динамічний фон). У сучасній науковій літературі представлено широкий спектр архітектурних рішень, що охоплюють двоетапні детектори, одноетапні підходи на базі згорткових нейронних мереж (CNN) та трансформерні архітектури [11]. Порівняльний аналіз актуальних моделей є необхідним етапом для обґрунтування вибору оптимального архітектурного рішення під задачу ідентифікації НРК.

Двоетапні детектори на базі регіональних мереж, такі як Faster R-CNN та Cascade R-CNN [12], демонструють високу точність локалізації завдяки двопрхідній архітектурі. На першому етапі механізм регіональних пропозицій (Region Proposal Network) генерує тисячі потенційних кандидатів (anchor boxes)

на наявність об'єктів, а на другому – механізми на кшталт RoI Pooling або RoI Align витягують ознаки для уточнення їхнього класу та координат [13]. Хоча така архітектура добре справляється зі складними сценами та дрібними об'єктами, вона породжує суттєву обчислювальну надлишковість. Двопрохідна обробка унеможлиблює паралельне виконання багатьох операцій і критично збільшує затримку обробки одного кадру, що робить цей клас архітектур малоприсадибним для розгортання в системах реального часу.

Одноетапні детектори сімейства YOLO (зокрема YOLOv6 [15], YOLOv7 [16], YOLOv8 [14] та YOLOv9 [23]) здійснили прорив у швидкодії порівняно з двоетапними підходами, вирішуючи задачу локалізації та класифікації за один прохід мережі як задачу регресії. Завдяки використанню сучасних пірамідальних структур ознак (FPN, PANet), вони здатні швидко обробляти зображення на різних масштабах. Однак їхня структурна залежність від алгоритму придушення немаксимумів (NMS) під час постпроцесингу вносить непередбачувану алгоритмічну затримку. Ця затримка експоненційно зростає зі збільшенням щільності об'єктів у кадрі – типова ситуація при ідентифікації колони НРК [17]. Крім того, базовий механізм локальних згорток обмежує здатність моделі враховувати глобальний контекст сцени [18], що є критично важливим для розрізнення НРК та візуально схожої цивільної техніки на складному фоні [19].

Трансформерні детектори типу DETR та його вдосконалені похідні (Deformable DETR, DAB-DETR, DINO [20]) концептуально змінюють підхід, відмовляючись від генерації анкорів та алгоритму NMS. Замість цього вони використовують механізм двочасткового оптимального зіставлення, що усуває проблему дублювання детекцій та підвищує стабільність прогнозування. Проте оригінальний DETR страждає від вкрай повільної збіжності під час навчання та квадратичної обчислювальної складності механізму просторової уваги  $O(n^2)$ , де  $n$  – кількість пікселів на картах ознак [10]. Модель Deformable DETR частково розв'язала цю проблему, знизивши складність до лінійної  $O(n)$  за рахунок використання деформованих точок уваги (увага зосереджується лише на невеликій кількості ключових точок навколо еталонної), але її загальна

швидкодія все ще поступається моделям сімейства YOLO [21].

Архітектура RT-DETR, запропонована Zhao та ін. у 2023 році та офіційно представлена на конференції CVPR 2024, стала першою трансформерною моделлю детекції, що повноцінно задовольняє жорстким вимогам реального часу [11]. Розвиваючи ідеї Dynamic DETR [22], RT-DETR успішно вирішує одразу дві ключові проблеми: обчислювальне «вузьке місце» механізму уваги та залежність від NMS-постпроцесингу. Можливість використання різних опорних мереж (наприклад, ResNet або PPHGNetV2) робить її гнучкою для адаптації під конкретні апаратні обмеження обчислювача. Детальний порівняльний аналіз розглянутих архітектур за ключовими характеристиками узагальнено у таблиці 1.1. Для об'єктивного оцінювання використано метрику середньої середньої точності  $mAP@0.5:0.95$  (стандартний бенчмарк MS COCO [24]), швидкодію (FPS), кількість параметрів моделі, а також наявність необхідності у постпроцесингу NMS.

Таблиця 1.1 – Порівняльний аналіз архітектур нейромереж для детекції об'єктів

Модель	$mAP@0.5:0.95$ (COCO)	FPS (GPU T4)	Параметри (M)	NMS	Джерело
Faster R-CNN	37,4	~17	41,8	Так	[13]
YOLOv8m	50,2	~80	25,9	Так	[14]
<b>YOLOv8-L</b>	<b>52,9</b>	<b>~71</b>	<b>43,7</b>	<b>Так</b>	<b>[14]</b>
Deformable DETR	46,2	~25	40,1	Ні	[21]
RT-DETR-X	54,8	~74	67,0	Ні	[11]
<b>RT-DETR-L</b>	<b>53,0</b>	<b>~114</b>	<b>32,0</b>	<b>Ні</b>	<b>[11]</b>

Порівняльний аналіз (див. таблиця 1.1) демонструє, що двоетапні детектори, попри високу точність, не відповідають вимогам реального часу через

надмірну обчислювальну складність. Одноетапні YOLO-моделі забезпечують прийнятну швидкодію, проте структурно залежать від NMS-постпроцесингу та обмежені у здатності аналізувати глобальний контекст сцени. Архітектура RT-DETR та її розширення RT-DETRv2 [25] позбавлені обох зазначених недоліків та забезпечують порівнянну або вищу точність при збереженні швидкодії, достатньої для роботи в режимі реального часу [23]. Альтернативою серед одноетапних детекторів є YOLOv10 [2], що також реалізує наскрізну детекцію. Окремо варто зазначити розвиток детекторів з відкритим словником (open-vocabulary): Grounding DINO [26], YOLO-World [27] та OWL-ViT [28] дозволяють виявляти об'єкти за текстовим описом. Хоча для задачі однокласової детекції НРК такий підхід не є необхідним, він може бути корисним на етапі масштабування. RT-DETR-L демонструє найкраще співвідношення точності, швидкодії та обчислювальних витрат, що робить її найбільш придатною як базову архітектуру для систем ідентифікації НРК.

Таким чином, аналіз існуючих архітектур показує, що трансформерний підхід загалом та архітектура RT-DETR зокрема є найбільш перспективним напрямом для розв'язання задачі ідентифікації НРК: він поєднує глобальний аналіз контексту сцени, відмову від NMS та достатню швидкодію для практичного застосування [23].

#### **1.4 Аналіз існуючих програмних рішень**

Аналіз програмних рішень доцільно відокремити від аналізу наукових підходів, оскільки на практиці важливими є не лише архітектурні принципи моделі, а й доступність інструментів для навчання, валідації, інференсу та інтеграції у прикладний застосунок. Для задачі ідентифікації НРК програмне рішення повинно підтримувати роботу з однокласовим датасетом, форматом YOLO-розмітки, GPU-прискоренням, автоматичним розрахунком метрик і збереженням навчених ваг.

Поширеним відкритим інструментом є екосистема Ultralytics, яка підтримує навчання та інференс моделей YOLO і RT-DETR, роботу з

конфігураційним файлом `data.yaml`, автоматичне формування графіків, матриць сплутаності та чекпоінтів `best.pt` і `last.pt` [29]. Її перевагою є швидке розгортання експериментального конвеєру мовою Python, а недоліком – залежність від готових реалізацій та необхідність уважного контролю параметрів навчання для трансформерних моделей. Зокрема, робота з архітектурами на основі механізму уваги, такими як RT-DETR, вимагає ретельного підбору швидкості навчання та стратегій регуляризації для запобігання перенавчанню.

Платформа Roboflow використовується для зберігання, підготовки та експорту датасетів у форматах, сумісних із сучасними детекторами об'єктів [30]. Серед інших джерел даних слід відзначити Objects365 [31] – масштабний набір для детекції, проте він не містить спеціалізованого класу НРК. Для цієї роботи перевагою Roboflow є наявність відкритого набору Small UGV detection з одним класом `ugv`, що відповідає постановці задачі. Водночас Roboflow не замінює власної експериментальної перевірки, оскільки якість кінцевого методу визначається не лише джерелом даних, а й параметрами навчання, стабільністю моделі та результатами тестування.

Альтернативними інструментами є PaddleDetection та NVIDIA TAO Toolkit. PaddleDetection надає широку колекцію моделей детекції, однак потребує окремого стеку PaddlePaddle, що ускладнює інтеграцію з Python/PyTorch-стеком. NVIDIA TAO Toolkit орієнтований на промислове навчання та оптимізацію моделей, проте є менш гнучким для навчально-дослідної роботи, де необхідно швидко змінювати код, аналізувати помилки та формувати власні графіки результатів.

З огляду на вимоги роботи, найбільш доцільним є використання відкритого Python-стеку: PyTorch, Ultralytics, OpenCV, NumPy, Pillow та Matplotlib. Така комбінація забезпечує повний цикл програмної реалізації: перевірку структури даних, навчання моделей, тестування, порівняння з `baseline`, візуалізацію результатів і створення графічного інтерфейсу оператора.

## 1.5 Мета та завдання роботи

Метою кваліфікаційної роботи є підвищення точності ідентифікації та швидкості виявлення наземних роботизованих комплексів на зображеннях шляхом розроблення нейромережевого методу.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз сучасного стану технологій та підходів до ідентифікації наземних роботизованих комплексів за допомогою методів комп'ютерного зору та глибокого навчання;

- формалізувати задачу ідентифікації наземних роботизованих комплексів як задачу детекції об'єктів на зображенні.

- розробити метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень, спрямований на підвищення точності та швидкості виявлення;

- виконати програмну реалізацію методу;

- провести експериментальне дослідження методу з використанням метрик точності та оцінкою швидкості виявлення.

## Розділ 2 Проектування нейромережевого методу ідентифікації наземних роботизованих комплексів за аналізом зображень

### 2.1 Концепція та етапи нейромережевого методу ідентифікації

Метод ґрунтується на формалізації задачі ідентифікації наземних роботизованих комплексів (НРК) у термінах теорії операторів. Розглянемо вхідний простір даних та математичний апарат, що описує перетворення зображення у структурований вектор ідентифікації об'єкта.

Нехай  $X$  – простір вхідних цифрових зображень, представлених у вигляді тривимірних тензорів:

$$X = \{I | I \in R^{H \times W \times 3}\} \quad (2.1)$$

де  $X$  – простір вхідних цифрових зображень;  $I$  – окреме RGB-зображення;  $H$  та  $W$  – висота і ширина зображення;  $3$  – кількість каналів колірного простору RGB.

Процес ідентифікації подається як послідовне перетворення вхідного зображення оператором попередньої обробки, нейромережевою моделлю та евристичною функцією фільтрації результатів:

$$R = H(M(D(I))) \quad (2.2)$$

де  $R$  – множина вихідних передбачень;  $H$  – евристична функція фільтрації результатів;  $M$  – нейромережева модель детекції;  $D$  – оператор попередньої обробки;  $I$  – вхідне зображення.

На етапі декодування запити-об'єкти взаємодіють із простором ознак через механізм перехресної уваги для локалізації цільових об'єктів. Результатом роботи методу є множина векторів ідентифікації, де кожен вектор описує окреме передбачення моделі:

$$r_i = (x_i, y_i, w_i, h_i, c_i, p_i), i = 1, \dots, n \quad (2.3)$$

де  $r_i$  – вектор  $i$ -го передбачення;  $x_i$  та  $y_i$  – координати центра обмежувальної рамки;  $w_i$  та  $h_i$  – ширина і висота рамки;  $c_i$  – ідентифікатор класу об'єкта;  $p_i$  – рівень достовірності передбачення;  $n$  – кількість передбачень, сформованих моделлю. Навчання моделі виконується шляхом мінімізації багатокomпонентної функції втрат:

$$L = \lambda_{cls} \cdot L_{cls} + \lambda_{box} \cdot L_{box} + \lambda_{IoU} \cdot L_{IoU} \quad (2.4)$$

де  $L$  – загальна функція втрат;  $\lambda_{cls}$ ,  $\lambda_{box}$  та  $\lambda_{IoU}$  – вагові коефіцієнти складових втрат;  $L_{cls}$  – класифікаційна складова;  $L_{box}$  – складова регресії координат рамки;  $L_{IoU}$  – складова просторового збігу прогнозованої та еталонної рамок. Така функція одночасно враховує правильність класу об'єкта і точність його локалізації.

## 2.2 Алгоритмічна модель нейромережевого методу

Узагальнену схему методу, що поєднує попередню обробку зображення, нейромережевий інференс та постаналіз результатів за шкалою достовірності  $H(p)$ , наведено на рисунку 2.1. Цей підхід дозволяє систематизувати процес проходження даних від моменту отримання первинного кадру сцени до формування фінального сповіщення із точними координатами виявленого об'єкта.

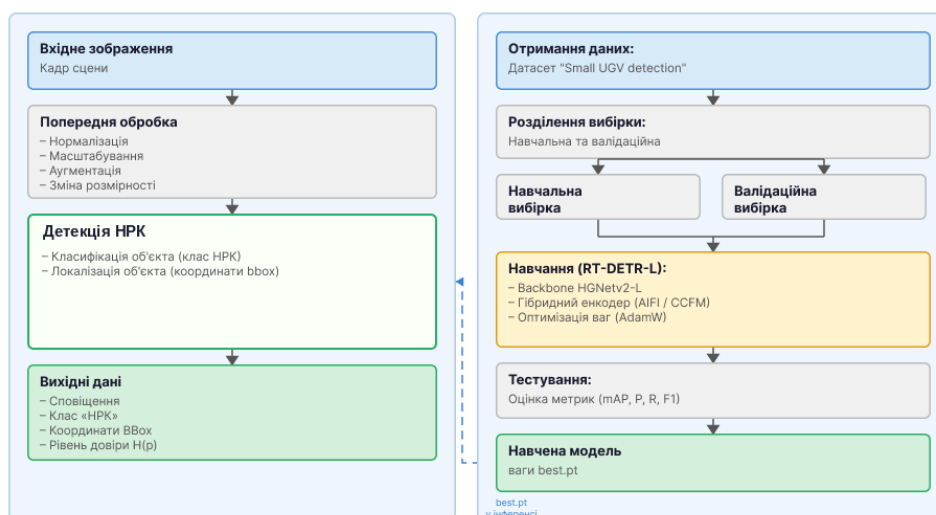


Рисунок 2.1 – Загальна схема нейромережевого методу ідентифікації НРК

Метод охоплює повний цикл обробки вхідного зображення: підготовку даних, нейромережевий інференс та постаналіз результатів. На етапі підготовки вхідне зображення приводиться до вхідного формату моделі із збереженням співвідношення сторін за допомогою letterbox-паддингу та нормалізується за значеннями каналів RGB. На етапі інференсу зображення проходить через гібридний енкодер RT-DETR-L, після чого трансформерний декодер формує фіксовану множину гіпотез щодо ймовірного розташування об'єктів [11]. На етапі постаналізу результати інференсу класифікуються за тривірневою шкалою достовірності  $H(p)$ : застосовується низький технічний поріг впевненості для збереження слабких гіпотез, а остаточний статус результату визначається після їх фільтрації за  $H(p)$ .

Нейромережева частина формує набір передбачень, а програмна евристика  $H(p)$  перетворює значення впевненості у прикладний статус результату: підтвержене виявлення, зона невизначеності або шум. Низький технічний поріг інференсу використовується для збереження слабких передбачень, але остаточний статус результату визначається після фільтрації за  $H(p)$ .

Для відтворюваності експерименту в методі явно задаються вхідний розмір зображення, спосіб масштабування, формат координат, пороги  $H(p)$ , поділ вибірки, модельні ваги та метрики оцінювання.



Рисунок 2.2 – Алгоритм роботи методу детекції

Загальний алгоритм функціонування методу включає такі кроки:

- Крок 1 – отримання вхідного зображення;
- Крок 2 – приведення зображення до вхідного формату із збереженням співвідношення сторін;
- Крок 3 – нормалізація значень інтенсивності пікселів у діапазоні [0, 1];
- Крок 4 – ієрархічне вилучення просторових ознак в опорному блоці та формування багаторівневих карт ознак;
- Крок 5 – аналіз контексту сцени механізмами самоуваги для врахування зв'язків між елементами об'єкта та фоном;
- Крок 6 – генерація запитів-гіпотез щодо ймовірного розташування об'єктів у просторі зображення;
- Крок 7 – нейромережевий інференс із технічним порогом  $\text{conf}=0.01$  для збереження слабких гіпотез;
- Крок 8 – перевірка гіпотези про наявності об'єктів (за відсутності об'єкта виводиться відповідне сповіщення оператору та припиняється обробка зображення, за його наявності виконуються наступні кроки )
- Крок 9 – пост-обробка зображення;
- Крок 10 – евристична фільтрація гіпотез;
- Крок 11 – класифікація результатів за трирівневою шкалою достовірності  $H(p)$ :

$$H(p) = C, p \geq 0.85; U, 0.50 \leq p < 0.85; L, 0.05 \leq p < 0.50; N, p < 0.05 \quad (2.5)$$

де  $H(p)$  – функція присвоєння статусу результату;  $p$  – значення впевненості, отримане після інференсу;  $C$  – підтвержена ідентифікація;  $U$  – зона невизначеності;  $L$  – низькоімовірна гіпотеза;  $N$  – шум; межі 0.85, 0.50 та 0.05 – пороги переходу між зонами достовірності.

– Крок 12 – отримання вихідного вектору результату: клас об'єкта, координати обмежувальної рамки та статус достовірності.

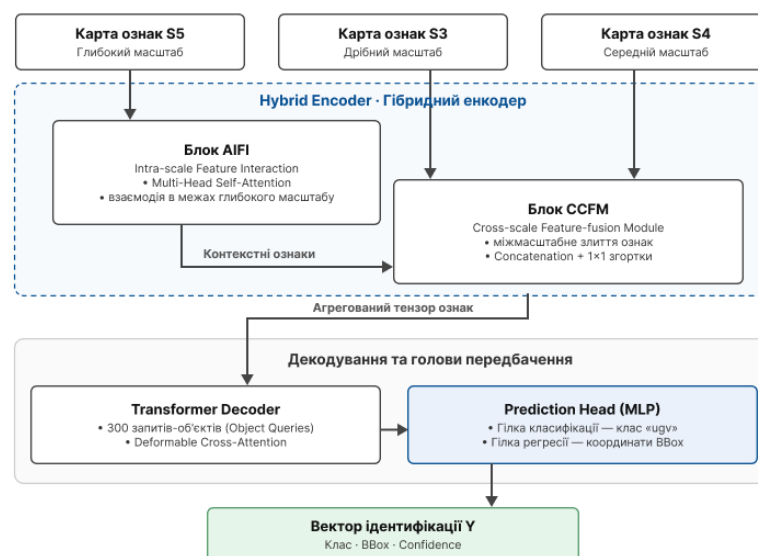
Алгоритм працює з одним або кількома передбаченнями на кадр. Якщо модель повертає декілька обмежувальних рамок, кожне передбачення

аналізується окремо, а для підсумкового повідомлення оператору використовується максимальне значення впевненості серед знайдених об'єктів. Така логіка відповідає прикладній задачі: якщо у кадрі є хоча б один НРК з високою ймовірністю, система повинна привернути увагу оператора, навіть якщо інші гіпотези мають нижчий рівень впевненості.

Параметр  $\text{conf}=0.01$  використовується як технічний механізм отримання широкого набору гіпотез від моделі. Застосування  $H(p)$  відокремлює сире неймережеве передбачення від підсумкового повідомлення оператору: модель формує гіпотези, а програмний блок постаналізу надає їм прикладний статус.

### 2.3 Архітектура неймережі

У методі використано архітектуру RT-DETR-L, що поєднує згорткові шари та трансформерні блоки. Конфігурація L (велика) містить приблизно 32 млн параметрів і використовує опорну мережу HGNetv2-L (альтернативою є опорна мережа ViT [37]). Модель виконує наскрізну детекцію без алгоритму немаксимального придушення (NMS), а її структура передбачає вилучення ієрархічних ознак, їх багатомасштабне злиття та подальше декодування запитів-об'єктів, що наочно зображено на рисунку 2.3.



Рисунк 2.3 – Архітектура неймережі RT-DETR-L

Центральним компонентом архітектури є гібридний енкодер, який виконує злиття багатомасштабних ознак. Блок включає AIFI (внутрішньомасштабна взаємодія ознак) та CCFM (міжмасштабний модуль злиття ознак).

AIFI використовує механізм багатоголової самоуваги для обробки карти ознак одного масштабу та формування контекстних зв'язків між віддаленими ділянками зображення.

CCFM здійснює агрегацію ознак різних рівнів, використовуючи операції об'єднання та шари  $1 \times 1$  згортки для вирівнювання розмірності каналів.

Блок декодера трансформера отримує на вхід 300 запитів-об'єктів. Ідентифікація виконується за механізмом деформованої перехресної уваги, у якому кожен запит аналізує обмежений набір релевантних точок на картах ознак.

Фінальна обробка здійснюється у блоці передбачень (Prediction Head), який складається з незалежних гілок багат шарових перцептронів (MLP). Гілка класифікації визначає ймовірність належності об'єкта до класу НРК, а гілка регресії обчислює координати обмежувальної рамки.

Теоретичним підґрунтям механізму багатоголової самоуваги є представлення уваги у вигляді зваженої суми значень залежно від скалярного добутку запитів та ключів [11]. В RT-DETR-L цей механізм подано у деформованому варіанті: кожен запит взаємодіє з обмеженим набором релевантних точок на карті ознак, а обчислювальна складність знижується з квадратичної  $O(n^2)$  до лінійної  $O(n)$  відносно розміру вхідного зображення [21]. В опорній мережі застосовуються залишкові зв'язки, які використовуються для стабілізації навчання багат шарових мереж. Для прискорення збіжності трансформерних детекторів використовується механізм шумозаглушення запитів, описаний у DN-DETR [38], а також методика неспрямованого попереднього навчання DETReg [39].

RT-DETR-L розглядає сцену як цілісну структуру, а не тільки як набір локальних ознак. Для малогабаритних НРК це означає, що модель опрацьовує просторово рознесені елементи об'єкта: корпус, колеса або гусениці, верхні

модулі, маркери, антени чи інші виступаючі частини. Механізм уваги використовується для встановлення зв'язків між такими фрагментами та формування гіпотези про цілісний об'єкт.

На рівні опорної мережі модель отримує багатомасштабне представлення сцени. Ознаки з малим кроком дискретизації корисні для локалізації меж об'єкта, а глибші ознаки з більшим кроком містять більш абстрактну інформацію про форму та контекст. Гібридний енкодер RT-DETR-L поєднує ці представлення, тому модель може одночасно враховувати дрібні контурні деталі та ширший фон, у якому знаходиться об'єкт. Це особливо важливо для зображень із плиткою, лініями підлоги, тінями або іншими регулярними візуальними структурами, які можуть створювати хибні локальні патерни.

Порівняно з класичними одноетапними детекторами, RT-DETR-L має іншу логіку формування передбачень. Набір запитів-об'єктів виконує роль обмеженої кількості гіпотез, які уточнюються декодером. За рахунок цього модель не потребує стандартної NMS-постобробки для усунення дублюючих рамок, що зменшує залежність затримки від кількості потенційних об'єктів у кадрі. У практичній системі це підвищує передбачуваність часу обробки, що важливо для застосування у відеопотоці. Адаптація та інтеграція базової архітектури RT-DETR-L у межах розробленого методу охоплює оптимізацію конвеєра підготовки даних, налаштування параметрів навчання, формування порогової шкали  $H(p)$ , а також порівняльний аналіз із baseline-моделями.

Характеристики нейромережевих моделей, використаних у роботі, суттєво відрізняються за архітектурними принципами та поведінкою у задачах детекції НРК. RT-DETR-L: 31,99 млн параметрів, 103,4 GFLOPs, наскрізна детекція без NMS, механізм деформованої перехресної уваги із 300 запитами-об'єктами, час інференсу 17,1 мс / зображення. Модель генерує фіксовану кількість гіпотез і відфільтровує їх внутрішньо; завдяки цьому при щільних сценах (колона НРК, декілька об'єктів в кадрі) кількість виведених рамок є передбачуваною. Водночас специфіка архітектури DETR-типу може проявлятися у надмірному накладанні обмежувальних рамок на однорідних фонах або в умовах часткового перекриття:

модель здатна генерувати дублюючі рамки на один і той самий об'єкт або виводити зайві рамки на текстурований фон, оскільки запити-об'єкти конкурують без жорсткого придушення. YOLOv8-L: 43,61 млн параметрів, 164,8 GFLOPs, одноетапний CNN-детектор із NMS-постобробкою, час інференсу 20,6 мс / зображення. Завдяки NMS модель стабільно усуває дублікати, проте за умови великої щільності об'єктів NMS може агресивно відкидати реальні сусідні цілі (придушення за перекриттям IoU). На практиці YOLO схильний до пропусків при сильному перекритті об'єктів, тоді як DETR краще зберігає повноту виявлення, але може малювати зайві рамки.

Різниця у поведінці визначається архітектурою: RT-DETR-L виконує детекцію через конкуренцію 300 запитів без постобробки – це означає, що модель може одночасно “бачити” один і той самий об'єкт кількома запитами, якщо їх не усуває механізм двочасткового зіставлення (угорський алгоритм). На текстурованих або однорідних фонах це може проявлятися як кілька накладених рамок поруч. YOLOv8-L, навпаки, формує прогнози через сітку беззаякорних точок і усуває дублікати NMS за IoU-порогом: при нормальних умовах результат є чистим і без дублів, але при IoU між сусідніми НПК вище порогу NMS – реальний об'єкт може бути придушений як “дублікат”. Для задачі з одним класом та помірною щільністю об'єктів обидві моделі дають прийнятні результати, але RT-DETR-L структурно краще підходить для сцен із перекриттями [40], а YOLOv8-L – для сцен із чистим фоном та малою кількістю об'єктів.

Для порівняльного аналізу використовується архітектура YOLOv8-L – одноетапний CNN-детектор із блоком CSPDarknet як опорною мережею, шиєю PAN-FPN для багатомасштабного злиття ознак (на відміну від однорівневих підходів, таких як YOLOF [41]) та трьома незалежними головами детекції. На відміну від RT-DETR-L, YOLOv8-L застосовує NMS-постобробку для усунення дублюючих рамок. Архітектуру YOLOv8-L наведено на рисунку 2.4.

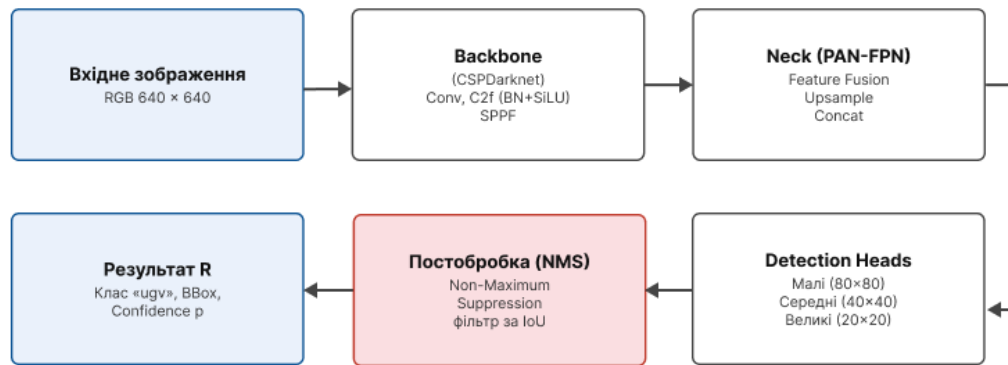


Рисунок 2.4 – Архітектура неймережі YOLOv8-L

## 2.4 Алгоритм навчання нейронної мережі

Алгоритм навчання нейронної мережі у роботі побудовано як послідовність контрольованих етапів, що забезпечують відтворюваність експерименту та стабільну збіжність моделі. На першому етапі виконується перевірка структури датасету: наявність каталогів train, val і test, відповідність зображень файлам розмітки, коректність кількості полів у YOLO-розмітці та належність координат діапазону  $[0, 1]$ .

На другому етапі завантажуються попередньо навчені ваги RT-DETR-L, після чого модель адаптується до цільового класу *ugv*. Вхідні зображення масштабуються до  $640 \times 640$  пікселів, а навчання виконується методом донавчання із використанням оптимізатора AdamW [42]. Для підвищення стабільності застосовується мала швидкість навчання, розігрів на початкових епохах і відключення AMP-режиму для RT-DETR-L.

На третьому етапі після кожної епохи здійснюється перевірка на валідаційній вибірці. За результатами валідації обирається найкращий чекпоінт *best.pt*, а механізм ранньої зупинки припиняє навчання, якщо протягом заданої кількості епох не спостерігається покращення основної метрики. Фінальною моделлю вважається чекпоінт із найкращим валідаційним результатом, а не обов'язково остання епоха навчання.

Фінальним етапом є тестування найкращого чекпоінта на незалежній тестовій вибірці. Як незалежний тестовий набір використовувались зображення

з репозиторію UGV-KPNet [43], зібраного для задачі детекції ключових точок малогабаритних наземних роботів (Li et al., IET Image Processing, 2021). Тестові дані не використовуються для підбору гіперпараметрів або вибору епохи, тому отримані показники Precision, Recall, F1-score, mAP та часу інференсу розглядаються як підсумкова оцінка працездатності розробленого методу.

## 2.5 Інформаційна модель вхідних даних

Інформаційна модель вхідних даних визначає, які саме дані надходять до методу, у якому форматі вони зберігаються та як перетворюються перед подачею до нейронної мережі. У межах роботи вхідним об'єктом є RGB-зображення  $I$ , а відповідною цільовою інформацією є множина анотацій  $Y$  для одного класу  $ugv$ .

Один навчальний приклад подається парою  $(I, Y)$ , де  $I$  містить матрицю пікселів із трьома каналами RGB, а  $Y$  містить нуль, одну або кілька анотацій об'єктів. Кожна анотація має формат YOLO:  $c, x_c, y_c, w, h$ , де  $c$  – ідентифікатор класу,  $x_c$  та  $y_c$  – нормалізовані координати центра рамки,  $w$  і  $h$  – нормалізовані ширина та висота обмежувальної рамки.

Нормалізація координат до діапазону  $[0, 1]$  дає змогу працювати із зображеннями різної початкової роздільної здатності. Перед інференсом кадр приводиться до розміру  $640 \times 640$  пікселів із збереженням співвідношення сторін, а значення інтенсивності пікселів перетворюються до числового діапазону, сумісного з нейромережевою моделлю.

Фонові зображення, у яких відсутній об'єкт класу  $ugv$ , також є частиною інформаційної моделі. Вони можуть мати порожній файл розмітки або не містити рядків з анотаціями. Такі приклади потрібні для перевірки здатності моделі не формувати хибнопозитивні спрацювання там, де цільового об'єкта немає.

Вихідний результат для кожного зображення формується як множина передбачень, що включають координати обмежувальної рамки, клас об'єкта та рівень впевненості. Цей формат використовується у математичній моделі методу, датасеті та програмній реалізації інтерфейсу оператора.

## 2.6 Характеристика датасету

Експериментальна частина роботи базується на відкритому наборі даних Small UGV detection, розміщеному на платформі Roboflow Universe. Згідно з описом сторінки датасету, його автором є MecatrnicUGV, тип задачі – Object Detection, кількість зображень – 5000, кількість класів – один клас *ugv* [30].

Зміст датасету відповідає задачі виявлення малогабаритних наземних роботизованих платформ на зображеннях. Для кожного об'єкта цільового класу задається обмежувальна рамка, тому під час оцінювання враховуються і факт виявлення НРК, і точність просторової локалізації. Для цього використовуються метрики Precision, Recall, F1-score, mAP@0.5 та mAP@0.5:0.95.

У фактичному експерименті використано структуру даних, сумісну з Ultralytics: окремі каталоги *images/* та *labels/*, а також конфігураційний файл *data.yaml* із шляхами до *train* і *val*. Після перевірки датасету навчальна частина містила 3803 зображення, валідаційна – 869 зображень із 868 анотованими об'єктами. У валідаційній вибірці зафіксовано 10 фонових зображень без об'єкта класу *ugv*. Для незалежного тестування використовувався окремий набір зображень з репозиторію UGV-KPNet [43], який не перетинається з даними Small UGV detection і не бере участі ані в навчанні, ані у виборі чекпоінта.

Перед запуском навчання виконувалася перевірка коректності розмітки: наявність зображень і відповідних файлів розмітки, допустимість ідентифікатора класу, п'ять числових значень у кожному рядку YOLO-розмітки та належність нормалізованих координат діапазону [0, 1]. Це важливо, оскільки помилки у розмітці можуть призвести до некоректного обчислення втрат локалізації та нестабільності навчання.

Обраний датасет використовується як однокласовий набір для виявлення об'єктів класу *ugv*. Такий формат відповідає постановці роботи: метод визначає наявність НРК у кадрі та формує обмежувальну рамку, а не виконує багатокласову класифікацію різних типів техніки.

Водночас однокласова структура датасету має певні обмеження, які враховуються під час інтерпретації результатів. Високі значення Precision і Recall на такому наборі не означають, що модель без додаткового навчання буде однаково добре розрізняти всі можливі типи наземних роботизованих платформ або всі види цивільної техніки. Отримані результати слід розглядати як оцінку в межах конкретного набору даних і як основу для подальшого розширення вибірки.

У валідаційній та тестовій вибірках наявні окремі фонові зображення. Вони використовуються для перевірки поведінки моделі в ситуаціях, коли цільового об'єкта немає, та для фіксації хибнопозитивних спрацювань.

Додатково датасет дає можливість перевірити якість просторової локалізації. Оскільки кожен цільовий об'єкт описується обмежувальною рамкою, можна оцінювати не лише факт виявлення, а й точність рамки за різними порогами IoU. Саме це обґрунтовує використання mAP@0.5:0.95 як суворішого показника порівняно з mAP@0.5. Для малогабаритного об'єкта навіть невелике зміщення рамки може помітно вплинути на якість подальшої інтерпретації сцени.

## 2.7 Метрики оцінювання

Для об'єктивної оцінки якості розробленого методу та порівняння його з існуючими рішеннями використовується система стандартних метрик детекції об'єктів. Вибір показників обумовлений необхідністю одночасного контролю здатності моделі до правильної класифікації та точності просторової локалізації НРК на зображенні.

Основними метриками в межах дослідження визначено такі показники:

- **точність (Precision)** – відображає частку істинно позитивних спрацювань серед усіх випадків, які модель класифікувала як НРК;
- **повнота (Recall)** – визначає здатність методу виявляти всі наявні об'єкти цільового класу у вибірці;

– **середня точність (mAP)** – інтегральний показник, що обчислюється як площа під кривою Precision-Recall.

Математично Precision, Recall, F1-score, IoU та mAP визначаються через такі співвідношення:

$$\text{Precision} = TP / (TP + FP) \quad (2.6)$$

де Precision – точність позитивного рішення; TP – кількість істинно позитивних спрацювань; FP – кількість хибнопозитивних спрацювань.

$$\text{Recall} = TP / (TP + FN) \quad (2.7)$$

де Recall – повнота виявлення; TP – кількість істинно позитивних спрацювань; FN – кількість хибнонегативних пропусків.

$$F1 = 2 \cdot P \cdot R / (P + R) \quad (2.8)$$

де F1 – гармонічне середнє точності та повноти; P – значення Precision; R – значення Recall.

$$\text{IoU} = |Bp \cap Bg| / |Bp \cup Bg| \quad (2.9)$$

де IoU – коефіцієнт перетину над об'єднанням; Bp – прогнозована обмежувальна рамка; Bg – еталонна обмежувальна рамка;  $\cap$  – площа перетину рамок;  $\cup$  – площа їх об'єднання.

$$mAP@0.5:0.95 = (1/10) \cdot \sum AP_t \quad (2.10)$$

де  $mAP@0.5:0.95$  – усереднене значення Average Precision для десяти порогів IoU від 0.50 до 0.95;  $AP_t$  – середня точність при порозі IoU, що дорівнює t.

Ключовим показником для порівняльного аналізу є  $mAP@0.5:0.95$ . Він враховує якість накладання обмежувальних рамок при різних порогах IoU, тому використовується для оцінювання просторової точності локалізації. Показник F1-score застосовується як інтегральна характеристика балансу між Precision та Recall.

З теоретичного погляду, крива Precision-Recall (PR-крива), площа під якою і становить значення AP (Average Precision), є стандартним інструментом оцінювання детекторів об'єктів при різних значеннях порогу рішення [6]. Для задач із сильним дисбалансом класів PR-крива використовується частіше за ROC-криву, оскільки кількість фонових регіонів на зображенні може суттєво перевищувати кількість цільових об'єктів [6].

Метрика Intersection over Union (IoU), що лежить в основі обчислення mAP, визначає ступінь просторового збігу прогнозованої та еталонної обмежувальної рамки і використовується для оцінювання якості локалізації [44]. Розширений варіант GIoU [45] враховує також площу об'єднання рамок. Розширений набір метрик потрібен для окремого аналізу класифікації, локалізації та швидкодії методу.

У роботі метрики трактуються не ізольовано, а як взаємодоповнювальна система показників. Precision характеризує надійність позитивного рішення: чим він вищий, тим менша частка випадків, коли система повідомляє про НРК помилково. Recall характеризує повноту виявлення: чим він вищий, тим менша ймовірність пропуску реального об'єкта. F1-score використовується для узагальнення цих двох властивостей, але не замінює їх окремого аналізу, оскільки однакове значення F1 може відповідати різним співвідношенням Precision і Recall.

Для задачі НРК особливе значення має аналіз матриці сплутаності. У ній хибнопозитивні спрацювання показують, скільки разів модель позначила фон як цільовий об'єкт, а хибнонегативні – скільки реальних об'єктів було пропущено. Такий аналіз є більш інтерпретованим для прикладної системи, ніж тільки інтегральне значення mAP, оскільки він безпосередньо відповідає на питання, якого типу помилки переважають у моделі.

Крім показників якості, у роботі оцінюється швидкодія. Основним часовим показником є час інференсу одного зображення у мілісекундах. Для задачі аналізу статичних зображень саме цей показник є коректнішим, ніж FPS, оскільки він безпосередньо відображає затримку обробки одного кадру.

Окремо враховуються кількість параметрів і GFLOPs. Кількість параметрів характеризує обсяг моделі та вимоги до пам'яті, а GFLOPs приблизно відображає обсяг обчислень для одного проходу. У сукупності з фактичним часом інференсу ці показники дозволяють оцінити не тільки якість моделі, а й доцільність її застосування на доступному GPU-прискорювачі.

## 2.8 Висновки до розділу 2

Виконано математичну формалізацію задачі нейромережевої ідентифікації НРК за аналізом зображень. Метод включає попередню обробку зображення, нейромережевий інференс та пост-аналіз результатів. Задачу ідентифікації НРК формалізовано у термінах операторного перетворення  $R=H(M(D(I)))$ , а результатом є множина векторів ідентифікації.

Сформовано алгоритмічну модель методу яка включає 12 кроків та передбачає отримання вхідного зображення, його попередню обробку, нейромережевий інференс, пост-аналіз за шкалою достовірності. Запропоновано тривірневу евристичну шкалу достовірності  $H(p)$  з порогамі 0,85/0,50/0,05, що відповідає статусам C/U/L/N.

Для класифікації та локалізації об'єктів використовується нейромережеві архітектури RT-DETR-L та YOLO, навчання яких здійснюються із використанням комбінованої функції втрат, яка забезпечує високу точність локалізації НРК. Визначено конфігурацію RT-DETR-L (31,99 млн параметрів, опорна мережа HGNetv2-L, 300 запитів-об'єктів, наскрізна без NMS) та YOLOv8-L (43,61 млн параметрів, NMS-постобробка).

Описано алгоритм навчання нейронної мережі. Описано спеціалізований датасет Small UGV detection, формат вхідних даних, принцип формування анотацій та порядок підготовки вибірок для навчання та валідації нейромережі.

Визначено критерії точності та швидкості оцінювання ідентифікації НРК як систему метрик Precision, Recall, F1-score, mAP@0.5 та mAP@0.5:0.95, а також матрицю сплутаності і час інференсу.

## Розділ 3 Розробка та експериментальна перевірка методу нейромережевої ідентифікації НРК

### 3.1 Проектна архітектура програмної реалізації та функціональна взаємодія компонентів

Програмна реалізація методу нейромережевої ідентифікації НРК побудована за трирівневою структурою, яка включає рівень взаємодії з користувачем, рівень нейромережевого аналізу та рівень управління даними.

У прототипі користувач ініціює аналіз через графічний інтерфейс. Після вибору зображення шлях до файлу передається модулю інференсу, модуль детекції повертає список рамок і значень впевненості, а інтерфейс відображає ці дані у вигляді рамки на зображенні, числового відсотка та статусу за шкалою  $N(p)$ .

Архітектура програмної реалізації задає послідовність передавання даних між компонентами: від завантаження вхідного зображення до виконання інференсу, фільтрації результатів за  $N(p)$ , накладання обмежувальної рамки та відображення рівня впевненості в інтерфейсі.

Загальна структура програмної реалізації включає такі функціональні рівні:

- рівень взаємодії з користувачем: завантаження зображень, запуск аналізу, відображення зображення, рамок і текстових повідомлень про результат;
- рівень нейромережевого аналізу: попередня обробка зображення, інференс RT-DETR-L, додатковий запуск YOLOv8-L у режимі порівняння та фільтрація результатів за шкалою  $N(p)$ ;
- рівень управління даними: робота з файлами ваг, конфігураційними файлами, результатами експериментів і журналами запусків.

Для порівняльного режиму передбачено окремий запуск YOLOv8-L. Основною моделлю у програмній реалізації є RT-DETR-L, а YOLOv8-L використовується як baseline для зіставлення результатів у межах одного інтерфейсу.

Функціональну архітектуру програмної реалізації методу наведено на рисунку 3.1.

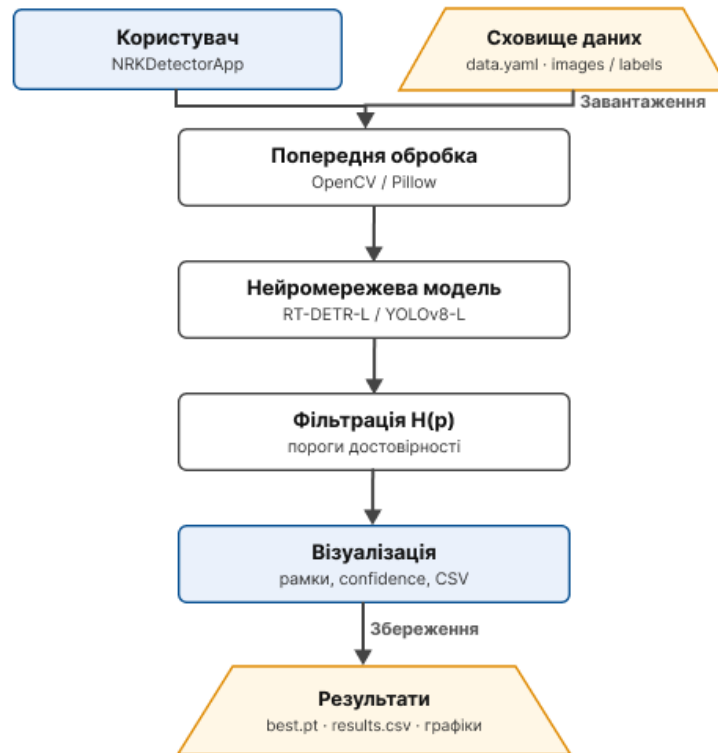


Рисунок 3.1 – Функціональна архітектура програмної реалізації методу неймережевої ідентифікації НРК

### 3.1.1 Особливості використання спеціалізованих програмних компонентів

Реалізація методу ідентифікації НРК використовує набір бібліотек і фреймворків для тензорних обчислень, роботи із зображеннями, навчання моделей, побудови графіків та відображення результатів у графічному інтерфейсі. Програмний стек побудовано в середовищі Python 3.10/3.11 з використанням PyTorch-сумісних інструментів.

До основних програмних компонентів реалізації належать:

- PyTorch та Ultralytics: завантаження попередньо навчених ваг, навчання, валідація, інференс RT-DETR-L і YOLOv8-L, збереження чекпоінтів best.pt і last.pt;
- NumPy: операції над багатовимірними масивами та проміжними числовими представленнями;

- OpenCV: читання зображень, зміна розміру, нормалізація та підготовка кадрів до подавання в нейромережеву модель;
- Matplotlib: побудова графіків метрик навчання, порівняльних діаграм і матеріалів для аналізу результатів;
- Pillow: відкриття зображень у графічному інтерфейсі, накладання рамок, масштабування області перегляду та збереження проміжних візуалізацій.

Графічний інтерфейс реалізовано класом NRKDetectorApp. Він відповідає за побудову вікна застосунку, розміщення елементів керування, обробку дій користувача та оновлення області візуалізації після завершення інференсу.

Клас ObjectDetector інкапсулює роботу з нейромережевими моделями. Він перевіряє наявність файлів best.pt, визначає тип моделі за шляхом, завантажує RT-DETR або YOLO через Ultralytics API та повертає узагальнений результат для графічного інтерфейсу.

Метод detect\_objects повертає словник із результатами RT-DETR-L і YOLOv8-L. Для кожної моделі зберігається максимальну впевненість та список координат рамок. Якщо режим YOLO вимкнений, інференс цієї моделі не запускається.

Клас HeuristicFilter відокремлює порогову логіку  $H(p)$  від інтерфейсу. Він отримує значення впевненості і повертає статус результату: підтверджена ідентифікація, зона невизначеності або шум.

### 3.1.2 Компоненти програмного комплексу

Програмний комплекс реалізовано мовою Python за модульним принципом. Його структура відокремлює підготовку даних, навчання моделей, інференс, валідацію, тестування та візуалізацію результатів.

Модуль NeuralNetworkTrainer запускає навчання RT-DETR-L і YOLOv8-L, передає параметри в Ultralytics API, контролює функції втрат, зберігає шляхи до чекпоінтів і запускає оцінювання. Модуль DatasetValidator перевіряє data.yaml та файли розмітки перед навчанням.

Модуль VisualizationAnalytics формує порівняльні графіки, CSV-звіти та зображення для експериментального аналізу. У результаті експерименту зберігаються PR-криві, F1-confidence, матриці сплутаності, приклади передбачень і порівняльні графіки метрик.

Графічний інтерфейс NRKDetectorApp реалізує завантаження зображення, запуск нейромережевого аналізу, відображення обмежувальної рамки та рівня впевненості. Результат додатково класифікується за шкалою  $H(p)$ .

Сценарій використання прототипу складається з вибору зображення у стандартному діалоговому вікні, запуску інференсу та відображення результату в одному робочому вікні. Для демонстрації роботи методу не потрібне ручне введення команд у терміналі. Для візуалізації результатів використовується бібліотека Pillow. Після отримання координат обмежувальної рамки зображення відкривається у форматі RGB, на нього накладаються рамки та підписи, після чого воно масштабується до області перегляду інтерфейсу.

Перемикач «Візуалізація (рамки)» розділяє числовий результат і графічне накладання рамок. Перемикач «Додатково: Аналіз YOLO» запускає порівняльний режим, у якому поряд із RT-DETR-L обчислюється результат YOLOv8-L.

У поточній реалізації шляхи до файлів ваг задані локально. Для перенесення застосунку на інше середовище ці шляхи можуть бути винесені до конфігураційного файлу.

### **3.2 Реалізація експериментального дослідження**

Для забезпечення об'єктивності результатів експеримент розділено на етапи: вибір оптимальних параметрів моделі здійснювався на валідаційній вибірці, а підсумкова перевірка ефективності методу – на незалежному тестовому наборі даних.

У межах експерименту порівнювалися дві моделі: запропонована RT-DETR-L та базова YOLOv8-L. Обидві моделі навчалися на тому самому наборі даних, з однаковим вхідним розміром  $640 \times 640$  пікселів і фіксованим  $seed=42$ . За

таких умов різниця в результатах пов'язується з архітектурою, параметрами оптимізації та особливостями збіжності, а не з різними даними.

Навчання та оцінювання виконувалися на платформі NVIDIA GeForce RTX 4060 Laptop GPU з 8 GB VRAM, CUDA 12.1, Python 3.11.14, PyTorch 2.5.1+cu121 та Ultralytics 8.4.46. Така конфігурація відповідає практичному сценарію використання методу на доступному GPU-прискорювачі, а не тільки на серверному обладнанні.

Перед запуском навчання виконувалася перевірка структури датасету. Скрипт контролював існування шляхів train, val і test, наявність зображень, відповідність файлів розмітки та коректність рядків розмітки. Попередній контроль цілісності даних дозволяє мінімізувати ризики переривання обчислювального процесу на етапі ініціалізації навчання. Для забезпечення числової стабільності обчислень та запобігання дивергенції градієнтів (виникнення невизначеностей типу NaN/Inf у функціях втрат) фінальний експеримент для моделі RT-DETR-L реалізовано з вимкненим режимом автоматичної змішаної точності (AMP). Додатково оптимізовано параметри збіжності: встановлено швидкість навчання на рівні 0.0001, тривалість етапу розігріву збільшено до 5 епох, а також застосовано помірні геометричні аугментації.

### **3.2.1 Навчання нейронної мережі та перевірка на валідаційній вибірці**

Навчання RT-DETR-L виконувалося методом тонкого налаштування попередньо навчених ваг rtdetr-l.pt. Для базового порівняння аналогічно використовувалася модель YOLOv8-L із попередньо навченими вагами yolov8l.pt. Навчальна вибірка застосовувалася для оновлення параметрів мережі, а валідаційна вибірка – для контролю якості після кожної епохи та вибору найкращої моделі.

Основні параметри експерименту: кількість епох – 100, розмір міні-батчу – 4, оптимізатор – AdamW, рання зупинка  $\text{patience}=20$ , вхідний розмір –  $640 \times 640$  пікселів. Для RT-DETR-L застосовано стабілізовані налаштування: швидкість навчання 0.0001, розігрів 5 епох, вимкнений AMP-режим та помірні геометричні аугментації (без агресивних політик типу AutoAugment [46]). Зазначена конфігурація параметрів забезпечила монотонну та стабільну збіжність функцій втрат без числових аномалій в обчисленнях.

RT-DETR-L зупинилася автоматично на 59-й епосі через механізм ранньої зупинки: протягом 20 епох не спостерігалось покращення основного валідаційного показника. Найкращий результат було отримано на 39-й епосі, після чого відповідний чекпоінт збережено як `best.pt`. YOLOv8-L завершила повний цикл із 100 епох; найкращий результат за  $\text{mAP}@0.5:0.95$  на валідаційній вибірці зафіксовано на 97-й епосі. Результати перевірки моделей на валідаційній вибірці наведено у таблиці 3.1.

Таблиця 3.1 – Результати перевірки моделей на валідаційній вибірці

Модель	Precision	Recall	F1-score	$\text{mAP}@0.5$	$\text{mAP}@0.5:0.95$	Час інференсу, мс
YOLOv8-L	0.981	0.996	0.988	0.991	0.888	20.0
RT-DETR-L	0.989	0.999	0.994	0.992	0.893	17.2

Валідаційні результати підтверджують стабільну збіжність обох моделей. RT-DETR-L отримала дещо вищі значення Precision, Recall, F1-score та  $\text{mAP}@0.5:0.95$ , а також менший час інференсу у фінальному валідаційному запуску. Водночас ці результати ще не є остаточною оцінкою методу, оскільки валідаційна вибірка використовувалася для вибору найкращого чекпоінта. Динаміку метрик  $\text{mAP}$  моделей на валідаційній вибірці наведено на рисунку 3.2.

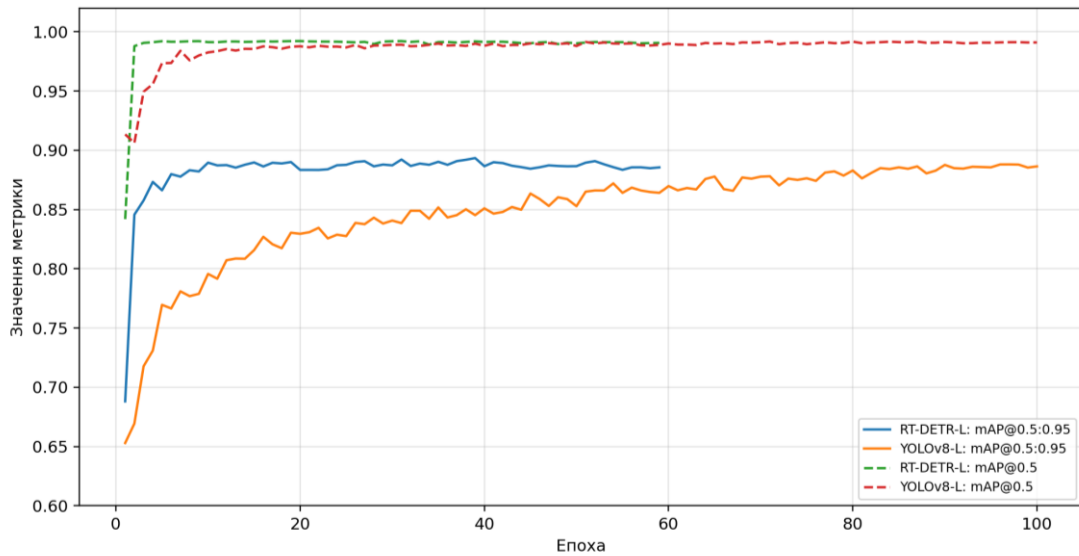


Рисунок 3.2 – Динаміка метрик mAP моделей на валідаційній вибірці

На рисунку 3.2 видно, що RT-DETR-L швидко досягає високих значень  $mAP@0.5$  та  $mAP@0.5:0.95$  вже на перших епохах, після чого переходить у режим повільного уточнення локалізації. YOLOv8-L демонструє плавнішу збіжність і поступово наближається до результатів RT-DETR-L, однак за підсумковим значенням  $mAP@0.5:0.95$  на валідаційній вибірці залишається нижче. Динаміку сумарних втрат RT-DETR-L під час навчання наведено на рисунку 3.3.

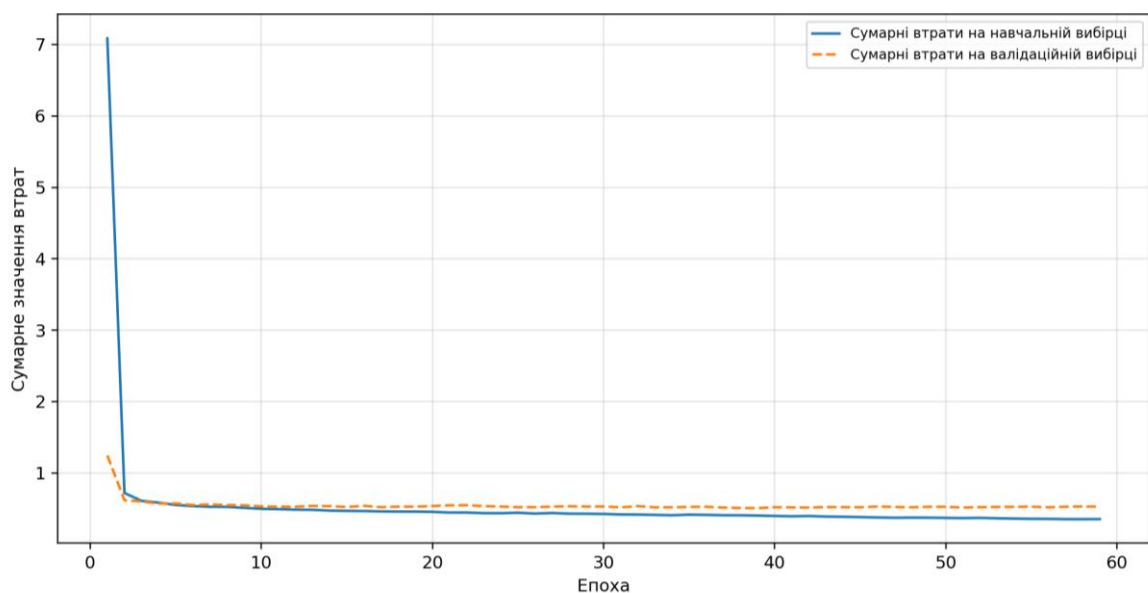


Рисунок 3.3 – Динаміка сумарних втрат RT-DETR-L під час навчання

Криві втрат RT-DETR-L показують різке зменшення помилки після першої епохи та подальше повільне уточнення параметрів. Відсутність NaN/Inf у `giou_loss`, `cls_loss` та `l1_loss` підтверджує, що стабілізований режим навчання придатний для отримання відтворюваного результату.

Збіжність RT-DETR-L має характер швидкої первинної адаптації з подальшим повільним уточненням. Уже після перших епох модель переходить від великих значень втрат до стабільного режиму, у якому зміни метрик стають невеликими. Така динаміка є очікуваною для донавчання попередньо навченої моделі: базові візуальні ознаки вже сформовані на великому наборі даних, а навчання на спеціалізованому датасеті уточнює їх під клас `ugv`.

Використання механізму ранньої зупинки з параметром `patience=20` забезпечило своєчасне припинення ітерацій за відсутності прогресу валідаційної метрики, що мінімізує ризик перенавчання та оптимізує витрати обчислювальних ресурсів. У цій роботі найкращий чекпоінт RT-DETR-L отримано на 39-й епосі, тоді як повний процес завершився на 59-й. Це дає змогу використовувати модель, яка показала найкращу якість на валідації, а не обов'язково останню модель навчання.

Для YOLOv8-L характерною була плавніша збіжність до кінця заданого циклу. Це свідчить про різницю в поведінці архітектур: CNN-базований детектор поступово уточнює локальні ознаки, тоді як RT-DETR-L швидше використовує попередньо сформоване глобальне представлення. Саме тому порівняння моделей у роботі виконується не за кількістю фактично пройдених епох, а за якістю їхніх найкращих чекпоінтів на однакових `validation` і `test` вибірках.

Динаміку сумарних втрат YOLOv8-L під час навчання наведено на рисунку 3.4.

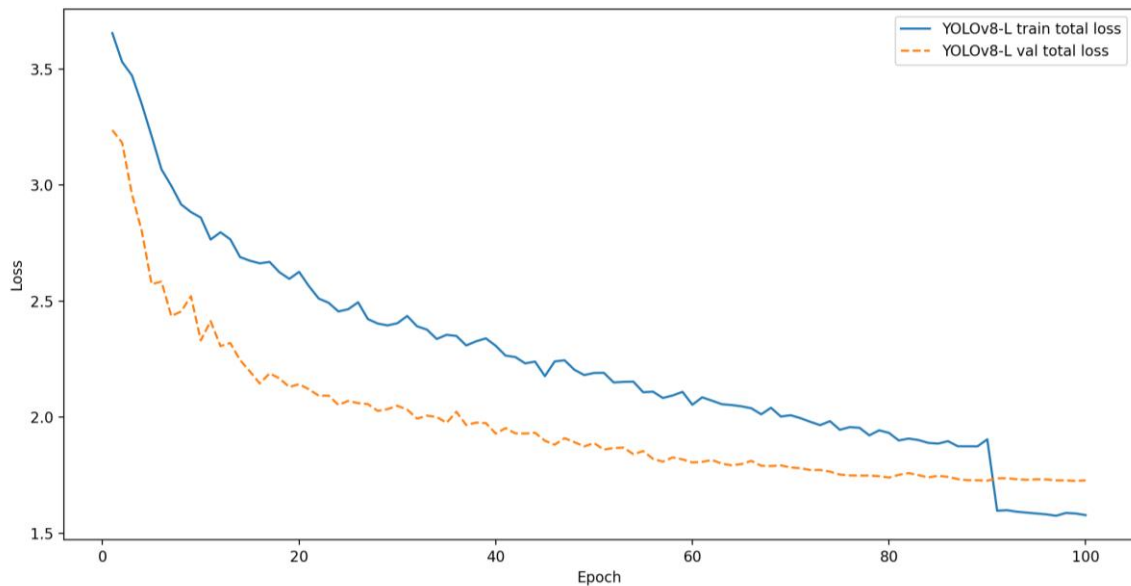


Рисунок 3.4 – Динаміка сумарних втрат YOLOv8-L під час навчання

Криві сумарних втрат YOLOv8-L демонструють монотонне спадання як на навчальній, так і на валідаційній вибірці: тренувальні втрати зменшуються приблизно з 3,65 до 1,58, а валідаційні – з 3,24 до 1,73. Стабільне зниження без розходження між train і val свідчить про відсутність перенавчання протягом усіх 100 епох. Помітне сходинкоподібне зниження тренувальних втрат на завершальних епохах пояснюється вимкненням мозаїчної аугментації (`close_mosaic`) наприкінці навчання, після чого модель донавчається на зображеннях без мозаїки, що наближає розподіл тренувальних даних до валідаційних.

На відміну від RT-DETR-L, який досягнув найкращого валідаційного результату вже на 39-й епосі та був зупинений механізмом ранньої зупинки на 59-й, YOLOv8-L поступово покращував показники впродовж усього циклу і досягнув найкращого значення  $mAP@0.5:0.95$  на валідаційній вибірці на 97-й епосі. Це підтверджує описану різницю в поведінці архітектур: трансформерний детектор швидше використовує попередньо сформоване глобальне представлення, тоді як CNN-базований детектор уточнює локальні ознаки повільніше, але стабільно.

### 3.2.2 Тестування методу

Після завершення навчання найкращий чекпоінт кожної моделі було протестовано на незалежному наборі даних UGV-KPNet [43], який містив 328 зображень, 325 анотованих об'єктів класу *ugv* та 3 фонові зображення без цільового об'єкта. Він не використовувався ні для оновлення ваг, ні для підбору гіперпараметрів, ні для вибору епохи зупинки.

Саме тестові результати використано для фінального порівняння RT-DETR-L з YOLOv8-L. Оцінювання виконувалося за однаковими метриками: Precision, Recall, F1-score,  $mAP@0.5$ ,  $mAP@0.5:0.95$  та час інференсу одного зображення. Порівняльну характеристику моделей на тестовій вибірці наведено у таблиці 3.2.

Таблиця 3.2 – Порівняльна характеристика моделей на тестовій вибірці

Модель	Precision	Recall	F1-score	$mAP@0.5$	$mAP@0.5:0.95$	Час інференсу, мс
YOLOv8-L	0.974	0.991	0.982	0.990	0.852	20.6
RT-DETR-L	0.991	1.000	0.995	0.992	0.863	17.1

На тестовій вибірці RT-DETR-L перевершила YOLOv8-L за всіма основними метриками: Precision, Recall, F1-score,  $mAP@0.5$ ,  $mAP@0.5:0.95$  та швидкістю інференсу. Особливо важливим є  $Recall = 1.000$ , тобто у межах тестового набору модель не пропустила жодного анотованого об'єкта класу *ugv*. Водночас  $Precision = 0.991$  свідчить про наявність лише поодиноких хибнопозитивних спрацювань. Такий баланс є цінним для системи моніторингу,

оскільки модель одночасно зменшує ризик пропуску цілі та не створює надмірного потоку помилкових сигналів.

Порівняння з YOLOv8-L використовується як baseline-оцінка. YOLOv8-L також демонструє високі результати, що підтверджує коректність експериментального конвеєру. У фінальному тестовому запуску RT-DETR-L має вищі значення основних метрик і менший час інференсу.

Результати тестування демонструють, що після стабілізації режиму навчання запропонована модель RT-DETR-L стала сильнішою не лише за критерієм надійності позитивного рішення, а й за повнотою виявлення. Порівняно з YOLOv8-L приріст RT-DETR-L становить +0.017 за Precision, +0.009 за Recall, приблизно +0.013 за F1-score, +0.002 за mAP@0.5 та +0.011 за mAP@0.5:0.95. Хоча абсолютна різниця в mAP невелика, вона отримана на незалежній тестовій вибірці та супроводжується кращою матрицею помилок.

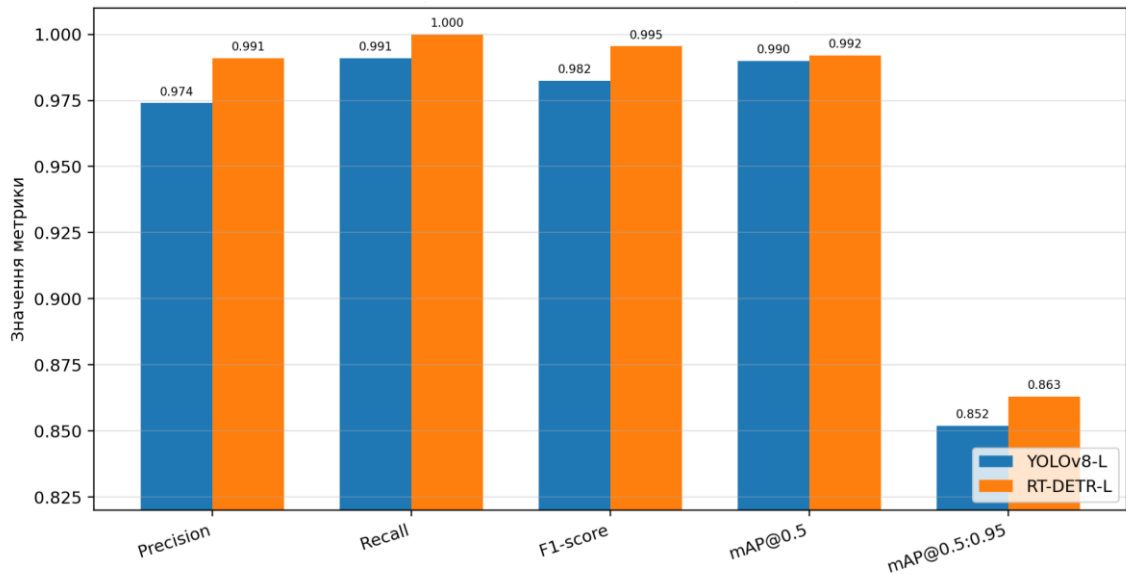


Рисунок 3.5 – Порівняння метрик RT-DETR-L та YOLOv8-L на тестовій вибірці

Діаграма на рисунку 3.5 показує, що обидві моделі досягли високої якості детекції, а RT-DETR-L має вищі значення за всіма п'ятьма порівнюваними метриками. Для задачі ідентифікації НРК окремо враховуються як пропуски

цільових об'єктів, так і хибнопозитивні спрацювання. Порівняння обчислювальної складності моделей наведено на рисунку 3.6.

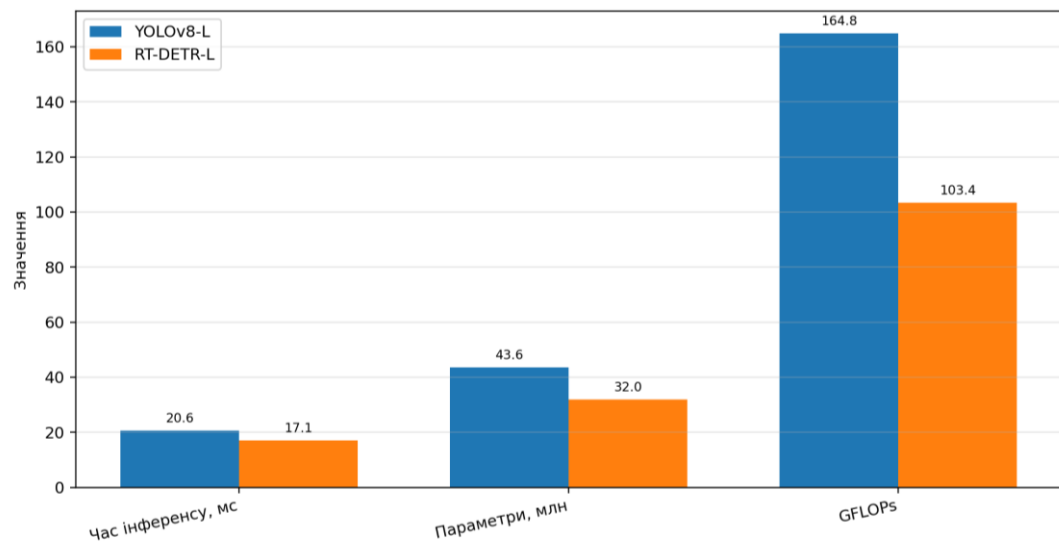


Рисунок 3.6 – Порівняння обчислювальної складності моделей

Окремо слід відзначити обчислювальну ефективність RT-DETR-L. Модель має приблизно 31,99 млн параметрів і 103,4 GFLOPs, тоді як YOLOv8-L має близько 43,61 млн параметрів і 164,8 GFLOPs. У фінальному тестовому запуску RT-DETR-L обробляла одне зображення за 17,1 мс, тоді як YOLOv8-L потребувала 20,6 мс. Отже, у цьому експерименті RT-DETR-L одночасно забезпечила кращу якість і меншу затримку інференсу. Матриці сплутаності моделей на тестовій вибірці наведено на рисунку 3.7.

Матриці сплутаності деталізують характер помилок. RT-DETR-L правильно виявила 325 об'єктів класу *ugv*, не сформувала хибнонегативних пропусків і дала 4 хибнопозитивні спрацювання на фоні. YOLOv8-L правильно виявила 322 об'єкти, пропустила 3 об'єкти та сформувала 13 хибнопозитивних спрацювань. Отримані результати відповідають меті роботи щодо підвищення надійності ідентифікації НРК.

Після корекції режиму навчання модель RT-DETR-L стабільно збігається, не формує NaN/Inf у *loss*-функціях, зберігає найкращий чекпоінт і демонструє високі показники на незалежній тестовій вибірці.

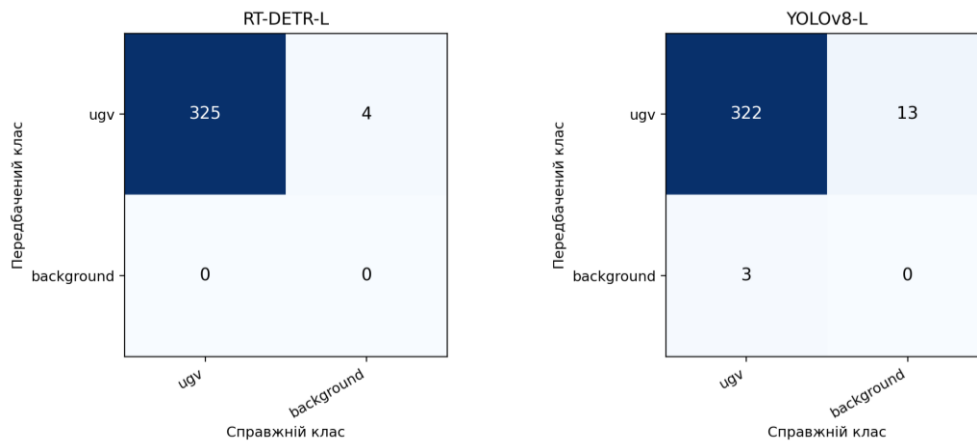


Рисунок 3.7 – Матриці сплутаності моделей на тестовій вибірці

Більш детальний аналіз помилок показує, що основна перевага RT-DETR-L проявляється не тільки в інтегральних метриках, а й у характері помилкових рішень. Для системи ідентифікації НРК важливо мінімізувати обидва типи помилок: хибнопозитивні спрацювання створюють помилкову тривогу, а хибнонегативні пропуски можуть призвести до втрати цільового об'єкта. За результатами тестування RT-DETR-L не сформувала хибнонегативних пропусків і мала менше хибнопозитивних спрацювань, ніж baseline.

Перевага за часом інференсу також має практичне значення. Різниця між 17,1 мс і 20,6 мс зменшує затримку обробки одного зображення, що важливо для подальшого розширення системи до послідовностей кадрів або роботи з додатковими модулями попередньої обробки.

Візуальні результати інтерфейсу підтверджують, що модель здатна працювати з реальними користувацькими сценаріями: завантаженим зображенням, масштабованим відображенням, накладанням обмежувальної рамки та порівнянням впевненості між моделями. Приклади роботи прототипу наведено на рисунках 3.8 та 3.9. Це важливо, оскільки дипломна робота передбачає не тільки навчання моделі, а й практичну реалізацію програмного комплексу, придатного для перевірки методу оператором.

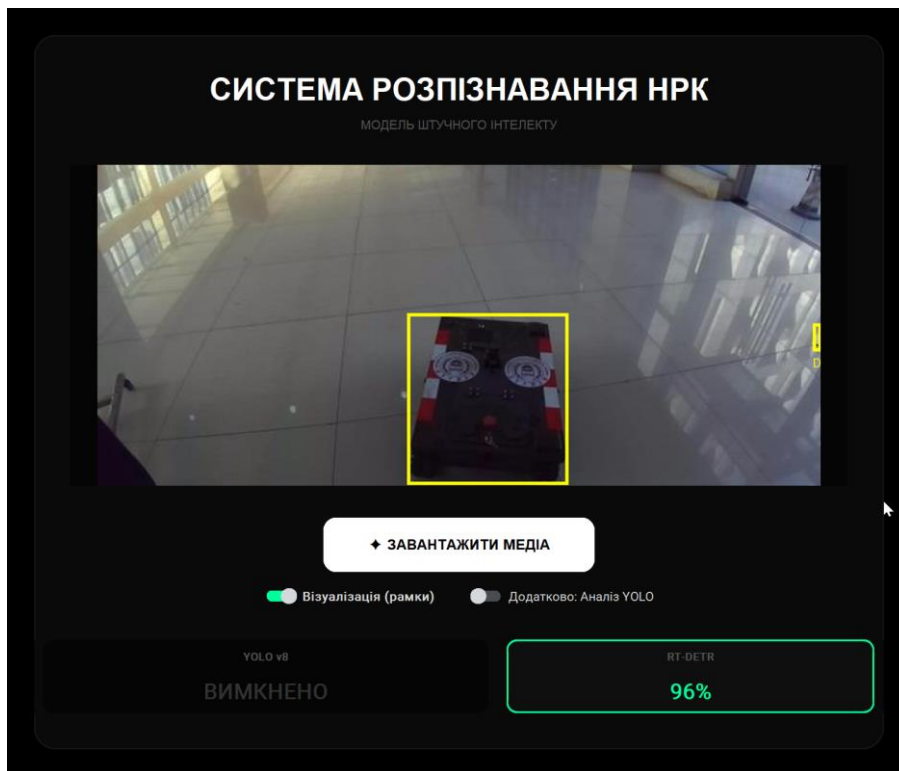


Рисунок 3.8 – Робота графічного інтерфейсу в режимі RT-DETR-L

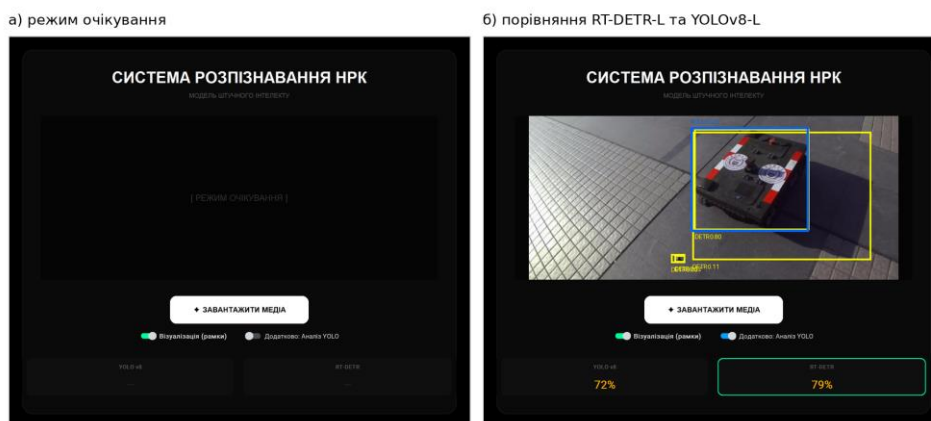


Рисунок 3.9 – Сценарії роботи графічного інтерфейсу застосунку для ідентифікації НРК

Головний користувацький сценарій функціонування інтерфейсу представлено на рисунку 3.8. Після завантаження вхідного файлу застосунок активує інференс моделі RT-DETR-L, формує обмежувальну рамку навколо знайденого об'єкта та обчислює відповідний коефіцієнт упевненості. Інші режими роботи програми (стан очікування, паралельне порівняння з baseline-моделлю YOLOv8-L, деактивація графічних контурів із збереженням числових

значень упевненості) наведено на рисунку 3.9. Створення таких варіантів відображення дозволяє оператору комплексно оцінювати достовірність детекції та гнучко варіювати візуальне навантаження на екранні форми.

Разом з тим, сформована нейромережева модель має певні обмеження і її не можна вважати інваріантною до будь-яких зовнішніх умов. Подальше вдосконалення методу доцільно спрямувати на диверсифікацію навчального датасету. Зокрема, шляхом додавання знімків з нестандартними ракурсами, складними типами фонів, змінними режимами штучного й природного освітлення, а також завад у вигляді диму чи рослинності. Проте у межах сформульованих вимог дослідження отримані показники свідчать про високу стабільність запропонованого підходу та його перевагу над архітектурою YOLOv8-L на тестовій вибірці.

Специфіка предметної області, пов'язана із закритістю реальних комплексів моніторингу, унеможлиблює використання загальнодоступних кількісних бенчмарків для порівняння ефективності системи з людиною-оператором. З огляду на пошуковий характер роботи, як референсну оцінку застосовано експертний метод: ту саму тестову вибірку незалежно розмічали два спостерігачі, що імітує детектування людиною-оператором.

Під час ручного опрацювання 328 тестових зображень люди систематично пропускали цільові об'єкти, якщо ті мали малі геометричні розміри або були частково перекриті (оклюзовані) фоновими елементами. Крім того, виникли суттєві розбіжності у суб'єктивному визначенні меж обмежувальних рамок.

Для кількісної оцінки експертного методу детекції, виконані кожним зі спостерігачів вручну, зіставлено з еталонною розміткою тестової вибірки (325 анотованих об'єктів класу *ugv*). За тими самими правилами, що й для нейромережевих моделей, обчислено кількість істинно позитивних виявлень (TP), пропусків (FN) і хибнопозитивних спрацювань (FP), а на їх основі – метрики Precision, Recall та F1-score. Узагальнені результати наведено в таблиці 3.3.

Таблиця 3.3 – Результати експертного методу (імітація детектування людиною-оператором)

Спостерігач	Precision	Recall	F1-score	Пропуски (FN)	Хибні (FP)	Час, с
Спостерігач 1	0,965	0,938	0,952	20	11	~34
Спостерігач 2	0,949	0,917	0,933	27	16	~29
<b>Усереднено</b>	<b>0,957</b>	<b>0,928</b>	<b>0,942</b>	–	–	<b>~31</b>

Усереднені показники експертного методу (Precision  $\approx$  0,957, Recall  $\approx$  0,928, F1-score  $\approx$  0,942) поступаються відповідним значенням обох нейромережових моделей. Це дає кількісну підставу стверджувати про підвищення точності ідентифікації НРК, визначене метою роботи: запропонований метод на основі RT-DETR-L перевищує рівень оператора як за повнотою виявлення (Recall = 1,000 проти  $\approx$  0,928), так і за надійністю позитивного рішення (Precision = 0,991 проти  $\approx$  0,957), а інтегральний F1-score зростає з  $\approx$  0,942 до 0,995. Базова модель YOLOv8-L (F1-score = 0,982) також перевищує рівень оператора, що підтверджує коректність експериментального конвеєра.

Обчислювальний експеримент засвідчив перевагу обох нейромережових архітектур над ручним аналізом. Модель RT-DETR-L забезпечила стовідсоткову повноту виявлення (Recall = 1,000) при 4 хибнопозитивних спрацюваннях на фоні, тоді як модель YOLOv8-L допустила 3 пропуски та 13 помилкових детекцій. На відміну від експертів, результати роботи RT-DETR-L є повністю детермінованими й відтворюваними.

Ключова перевага автоматизованого підходу полягає у високій швидкодії (інференс триває 17,1 мс на один кадр проти десятків секунд, необхідних людині) та незалежності від фактору втоми оператора. За умов обмеженого обсягу відкритих статистичних даних такий порівняльний підхід є методологічно обґрунтованим і достатнім для підтвердження ефективності розробленого програмного комплексу.

### 3.3 Висновки до розділу 3

Виконано програмну реалізацію методу нейромережевої ідентифікації НРК за аналізом зображень.

Архітектура програмної реалізації задає послідовність передавання даних між компонентами: від завантаження вхідного зображення до виконання інференсу, фільтрації результатів за  $N(p)$ , накладання обмежувальної рамки та відображення рівня впевненості в інтерфейсі.

Визначено загальну структуру програмної реалізації яка включає 3 функціональні рівні – рівень взаємодії з користувачем, рівень нейромережевого аналізу та рівень управління даними.

Реалізований програмний комплекс складається з модулів перевірки датасету, навчання моделей, інференсу, валідації, тестування, формування графіків і підготовки звітних матеріалів.

Для програмної реалізації було використано PyTorch та Ultralytics (навчання та інференс RT-DETR-L і YOLOv8-L), NumPy (операції над масивами), OpenCV (попередня обробка зображень), Matplotlib (побудова графіків та матриць сплутаності), Pillow (накладання рамок у графічному інтерфейсі), tkinter (графічний інтерфейс NRKDetectorApp).

Навчання RT-DETR-L було налаштовано на 100 епох, однак завершилося на 59-й епосі через механізм ранньої зупинки; найкращий чекпоінт отримано на 39-й епосі. Валідаційні метрики найкращої моделі становили Precision = 0.989, Recall = 0.999, F1-score = 0.994, mAP@0.5 = 0.992 та mAP@0.5:0.95 = 0.893. Це підтвердило стабільність навчання та доцільність використаних параметрів оптимізації.

На тестовій вибірці RT-DETR-L досягла Precision = 0.991, Recall = 1.000, F1-score = 0.995, mAP@0.5 = 0.992 та mAP@0.5:0.95 = 0.863. Порівняно з YOLOv8-L запропонована модель має вищі значення всіх основних тестових метрик, менше хибнопозитивних спрацювань, відсутність хибнонегативних

пропусків у матриці сплутаності та менший час інференсу – зокрема, приріст за  $mAP@0.5:0.95$  складає +1,3%, а перевага за часом інференсу – близько 17%.

Час інференсу RT-DETR-L у режимі  $640 \times 640$  становить 17,1 мс на зображення. У межах роботи метод застосовується до статичних зображень, проте отримане значення затримки теоретично відповідає  $\sim 58,5$  кадрів/с, що підтверджує достатній запас швидкодії для можливого майбутнього розширення методу на аналіз відеоряду. Експериментальні результати підтверджують практичну придатність розробленого методу та доцільність його інтеграції у прикладні інтерфейси оператора-спостерігача системи моніторингу наземної обстановки для Сил Оборони України.

## Загальні висновки

У процесі виконання кваліфікаційної роботи бакалавра було досягнуто мету, визначену на етапі постановки задачі, а саме – підвищення точності ідентифікації та швидкості виявлення НРК на зображеннях шляхом розроблення нейромережевого методу.

Для досягнення поставленої мети виконані такі задачі:

- проведено аналіз сучасного стану технологій та підходів до ідентифікації наземних роботизованих комплексів за допомогою методів комп'ютерного зору та глибокого навчання;

- формалізовано задачу ідентифікації наземних роботизованих комплексів як задачу детекції об'єктів на зображенні.

- розроблено метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень, спрямований на підвищення точності та швидкості виявлення;

- виконано програмну реалізацію методу;

- проведено експериментальне дослідження методу з використанням метрик точності та оцінкою швидкості виявлення.

Проаналізовано предметну область, класичні та сучасні підходи до детекції об'єктів, а також переваги й обмеження одноетапних CNN-детекторів і трансформерних архітектур. Встановлено, що для задачі ідентифікації НРК важливими є не тільки швидкість обробки кадру, а й стійкість до складного фону, оклюзій, хибнопозитивних спрацювань і пропусків цільових об'єктів.

Сформовано математичну та алгоритмічну модель методу. Задачу визначено як однокласову детекцію об'єкта *ugv*, описано послідовність перетворень вхідного зображення, у роботі було використано архітектури RT-DETR-L та YOLO і наведено інформаційну модель даних.

Описано алгоритм навчання нейронної мережі та описано датасет Small UGV detection.

Для оцінки розробленого методу визначено набір критерій, який включає Precision, Recall, F1-score, mAP@0.5 та mAP@0.5:0.95.

Реалізовано програмний комплекс і проведено експериментальну перевірку. Навчання виконувалося на навчальній вибірці з контролем якості на валідаційній вибірці, а фінальна оцінка проводилася окремо на незалежному тестовому наборі UGV-KPNet, що не перетинається з навчальними даними.

За результатами валідації найкраща модель RT-DETR-L досягла Precision = 0.989, Recall = 0.999, F1-score = 0.994, mAP@0.5 = 0.992 та mAP@0.5:0.95 = 0.893. Навчання RT-DETR-L було налаштовано на 100 епох, але завершилося на 59-й епосі через механізм ранньої зупинки; найкращий чекпоінт збережено на 39-й епосі. Це підтвердило стабільну збіжність моделі без появи NaN/Inf у функціях втрат.

Фінальне тестування на незалежній тестовій вибірці показало, що RT-DETR-L має Precision = 0.991, Recall = 1.000, F1-score = 0.995, mAP@0.5 = 0.992 та mAP@0.5:0.95 = 0.863. Базова YOLOv8-L на тих самих тестових даних отримала Precision = 0.974, Recall = 0.991, F1-score = 0.982, mAP@0.5 = 0.990 та mAP@0.5:0.95 = 0.852. Отже, запропонований метод перевищив baseline за всіма ключовими тестовими метриками: приріст за mAP@0.5:0.95 складає +1,3%, за Precision – +1,7%, за Recall – +0,9%, а час інференсу скорочено на 17%.

Матриці сплутаності показали, що RT-DETR-L правильно виявила 325 об'єктів класу *ugv*, не мала хибнонегативних пропусків і сформувала 4 хибнопозитивні спрацювання. Для YOLOv8-L зафіксовано 322 правильні виявлення, 3 пропуски та 13 хибнопозитивних спрацювань. Це підтверджує, що запропонований метод краще відповідає поставленій меті підвищення надійності ідентифікації НРК.

Практична цінність роботи полягає у створенні відтворюваного програмного конвеєру для перевірки датасету, навчання, валідації, тестування та порівняння моделей, а також у розробці прототипного застосунку з графічним інтерфейсом оператора. Отримані результати показали, що RT-DETR-L забезпечує час обробки одного статичного зображення 17,1 мс. Оскільки тема є

принципово новою і кількісних публічних бенчмарків порівняння з оператором не існує, проведено якісне зіставлення: при ручному перегляді тестової вибірки спостерігачі фіксували систематичні пропуски малогабаритних об'єктів та розбіжності у визначенні меж рамки. Нейромережевий метод позбавлений цих недоліків: результат детермінований, не залежить від стомлення та є повністю відтворюваним. Архітектурна відмінність моделей визначає тип помилок: RT-DETR-L схильна до надмірних рамок на однорідних фонах (4 FP), YOLOv8-L може придушувати сусідні цілі через NMS (3 пропуски, 13 FP). Подальший розвиток роботи доцільно спрямувати на розширення вибірки за рахунок різних умов зйомки та оптимізацію моделі для слабших обчислювальних платформ.

**Перелік посилань**

1. Brave1 [Електронний ресурс] : державний кластер оборонних технологій України. 2023. URL: <https://brave1.gov.ua> (дата звернення: 10.05.2026).
2. Wang A., Chen H., Liu L. et al. YOLOv10: Real-Time End-to-End Object Detection // *arXiv preprint*. 2024. URL: <https://arxiv.org/abs/2405.14458>.
3. Mao J., Shi S., Wang X. et al. 3D Object Detection for Autonomous Driving: A Comprehensive Survey // *International Journal of Computer Vision*. 2023. Vol. 131, No. 8. P. 1909–1963. DOI: 10.1007/s11263-023-01777-w.
4. Wu Y., Ding Y., Bian G. et al. A Survey of Object Detection for UAVs Based on Deep Learning // *Drones*. 2022. Vol. 6, Is. 3. P. 55. DOI: 10.3390/drones6030055.
5. Du D., Qi Y., Su H. et al. VisDrone-DET2023: The Vision Meets Drone Object Detection Challenge Results // *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2023. P. 2149–2158. DOI: 10.1109/ICCVW60793.2023.00220.
6. Zou Z., Chen K., Shi Z. et al. Object Detection in 20 Years: A Survey // *Proceedings of the IEEE*. 2023. Vol. 111, No. 3. P. 257–276. DOI: 10.1109/JPROC.2023.3238524.
7. Mumuni A., Mumuni F. Data augmentation: A comprehensive survey of modern approaches // *Array*. 2022. Vol. 16. P. 100258. DOI: 10.1016/j.array.2022.100258.
8. Khan S., Naseer M., Hayat M. et al. Transformers in Vision: A Survey // *ACM Computing Surveys*. 2022. Vol. 54, No. 10. P. 1–41. DOI: 10.1145/3505244.
9. Liu Z., Hu H., Lin Y. et al. Swin Transformer V2: Scaling Up Capacity and Resolution // *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. P. 12009–12019. DOI: 10.1109/CVPR52688.2022.01170.
10. Shehzadi T., Hashmi K. A., Stricker D., Afzal M. Z. Object Detection with Transformers: A Review // *arXiv preprint*. 2023. URL: <https://arxiv.org/abs/2306.04670>.

11. Zhao Y., Lv W., Xu S. et al. DETRs Beat YOLOs on Real-time Object Detection // 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2024. P. 16965–16974. DOI: 10.1109/CVPR52729.2024.01594
12. Cai Z., Vasconcelos N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. Vol. 43, No. 5. P. 1483–1498.
13. Padilla R., Passos W. L., Dias T. L. B. et al. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit // *Electronics*. 2021. Vol. 10, No. 3. P. 279. DOI: 10.3390/electronics10030279.
14. Jocher G., Chaurasia A., Qiu J. Ultralytics YOLOv8 // *GitHub repository*. 2023. URL: <https://github.com/ultralytics/ultralytics>.
15. Li C., Li L., Jiang H. et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications // *arXiv preprint*. 2022. URL: <https://arxiv.org/abs/2209.02976>.
16. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. P. 7464–7475. DOI: 10.1109/CVPR52729.2023.00721.
17. Tian Y., Zhao L., Chen Y. Detection and Tracking of Vehicles in Adverse Weather Using Deep Learning // *IEEE Transactions on Intelligent Transportation Systems*. 2023. Vol. 24. P. 6792–6804.
18. Song H., Sun D., Chun S. et al. ViDT: An Efficient and Effective Fully Transformer-based Object Detector // *International Conference on Learning Representations (ICLR)*. 2022. URL: <https://arxiv.org/abs/2110.03921>.
19. Terven J., Cordova-Esparza D. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS // *Machine Learning and Knowledge Extraction*. 2023. Vol. 5, No. 4. P. 1680–1716. DOI: 10.3390/make5040087.

20. Zhang H., Li F., Liu S. et al. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection // *International Conference on Learning Representations (ICLR)*. 2023. URL: <https://arxiv.org/abs/2203.03605>.
21. Zhu X., Su W., Lu L. et al. Deformable DETR: Deformable Transformers for End-to-End Object Detection // *International Conference on Learning Representations (ICLR)*. 2021. URL: <https://arxiv.org/abs/2010.04159>.
22. Dai X., Chen Y., Yang J. et al. Dynamic DETR: End-to-End Object Detection with Dynamic Attention // *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021. P. 2988–2997.
23. Wang C.-Y., Yeh I.-H., Liao H.-Y. M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information // *arXiv preprint*. 2024. URL: <https://arxiv.org/abs/2402.13616>.
24. Minaee S., Boykov Y., Porikli F. et al. Image Segmentation Using Deep Learning: A Survey // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022. Vol. 44, No. 7. P. 3523–3542. DOI: 10.1109/TPAMI.2021.3059906.
25. Lyu C., Zhang W., Huang H. et al. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer // *arXiv preprint*. 2024. URL: <https://arxiv.org/abs/2407.17140>.
26. Liu S., Zeng Z., Ren T. et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection // *arXiv preprint*. 2024. URL: <https://arxiv.org/abs/2303.05499>.
27. Cheng T., Song L., Ge Y. et al. YOLO-World: Real-Time Open-Vocabulary Object Detection // *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024. P. 16901–16911. URL: <https://arxiv.org/abs/2401.17270>.
28. Minderer M., Gritsenko A., Stone A. et al. Simple Open-Vocabulary Object Detection with Vision Transformers // *European Conference on Computer Vision (ECCV)*. 2022. DOI: 10.1007/978-3-031-19760-4\_40.

29. Jocher G., Chaurasia A., Qiu J. Ultralytics YOLO Docs. 2024. URL: <https://docs.ultralytics.com>.
30. MecatrnicUGV. Small UGV detection dataset // *Roboflow Universe*. 2024. URL: <https://universe.roboflow.com/mecatrnicaugv-hxnnv/small-ugv-detection>.
31. Shao S., Li Z., Zhang T. et al. Objects365: A Large-Scale, High-Quality Dataset for Object Detection // *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019. P. 8430–8439.
32. Ge Z., Liu S., Wang F. et al. YOLOX: Exceeding YOLO Series Detectors // *arXiv preprint*. 2022. URL: <https://arxiv.org/abs/2107.08430>.
33. Feng D., Haase-Schütz C., Rosenbaum L. et al. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges // *IEEE Transactions on Intelligent Transportation Systems*. 2022. Vol. 23, No. 2. P. 1341–1360. DOI: 10.1109/TITS.2020.3029528.
34. Oquab M., Darcet T., Moutakanni T. et al. DINOv2: Learning Robust Visual Features without Supervision // *arXiv preprint*. 2023. URL: <https://arxiv.org/abs/2304.07193>.
35. He K., Chen X., Xie S. et al. Masked Autoencoders Are Scalable Vision Learners // *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. P. 16000–16009. DOI: 10.1109/CVPR52688.2022.01553.
36. Kirillov A., Mintun E., Ravi N. et al. Segment Anything // *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023. P. 3992–4003. DOI: 10.1109/ICCV51070.2023.00371.
37. Li Y., Mao H., Girshick R. et al. Exploring Plain Vision Transformer Backbones for Object Detection // *European Conference on Computer Vision (ECCV)*. 2022. DOI: 10.1007/978-3-031-19760-4\_17.
38. Li F., Zhang H., Liu S. et al. DN-DETR: Accelerate DETR Training by Introducing Query DeNoising // *2022 IEEE/CVF Conference on Computer*

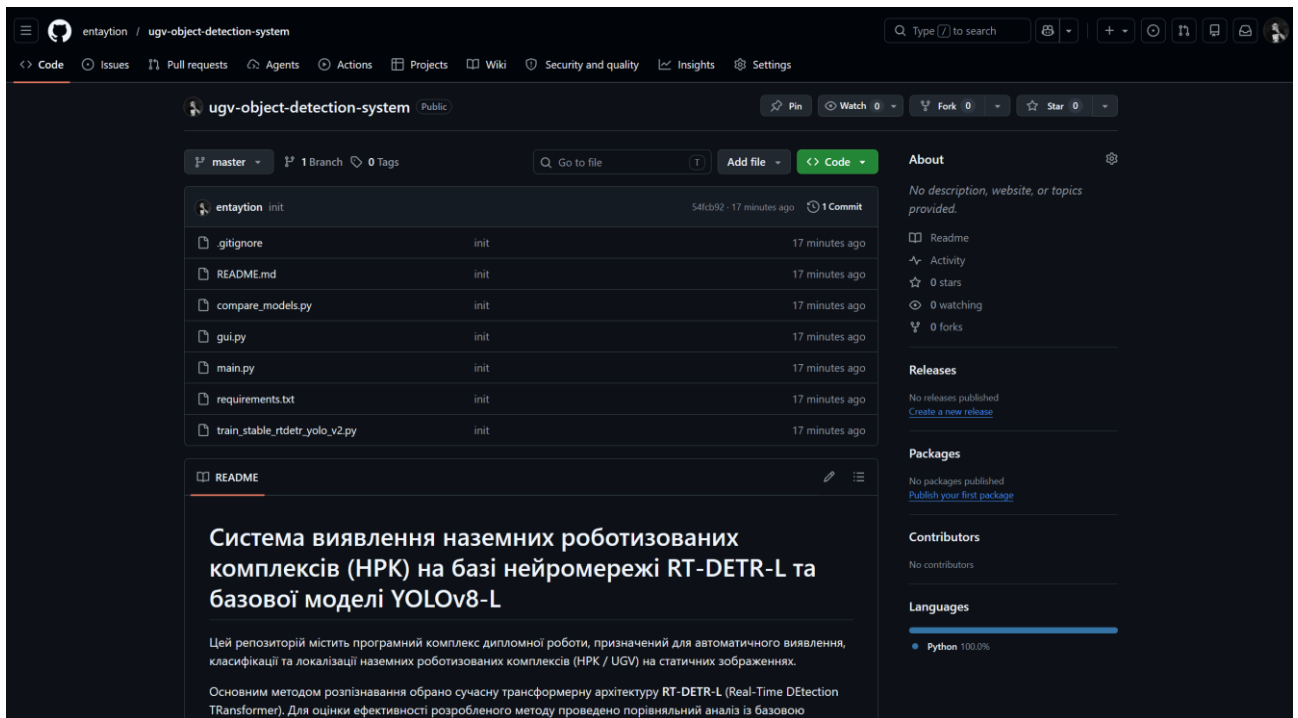
- Vision and Pattern Recognition (CVPR). 2022. P. 13619–13627. DOI: 10.1109/CVPR52688.2022.01325.
39. Bar A., Wang X., Kantorov V. et al. DETReg: Unsupervised Pretraining with Region Priors for Object Detection // *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. P. 14605–14614. URL: <https://arxiv.org/abs/2106.04550>.
  40. Xu T., Xu Z., Chen Z. et al. DAMO-YOLO: A Report on Real-Time Object Detection Design // *arXiv preprint*. 2022. URL: <https://arxiv.org/abs/2211.15444>.
  41. Chen Q., Wang Y., Yang T. et al. You Only Look One-level Feature // *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021. P. 13039–13048.
  42. PyTorch Foundation. PyTorch Documentation: torch.optim.AdamW. 2024. URL: <https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html>.
  43. Zafar M. H., Moosavi S. K. R., Sanfilippo F. Enhancing unmanned ground vehicle performance in search and rescue operations: integrated gesture-control and deep learning framework for optimised victim detection // *Frontiers in Robotics and AI*. 2024. Vol. 11. DOI: 10.3389/frobt.2024.1356345.
  44. Zheng Z., Wang P., Ren D. et al. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation // *IEEE Transactions on Cybernetics*. 2022. Vol. 52, No. 8. P. 8574–8586. DOI: 10.1109/TCYB.2021.3095305.
  45. Yu J., Jiang Y., Wang Z. et al. UnitBox: An Advanced Object Detection Network // *ACM International Conference on Multimedia*. 2016. P. 516–520.
  46. Cubuk E., Zoph B., Mane D. et al. AutoAugment: Learning Augmentation Strategies from Data // *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019. P. 113–123.

# ДОДАТКИ

## Додаток А

### Посилання на GitHub-репозиторій проєкту

Вихідний код програмної реалізації методу нейромережевої ідентифікації наземних роботизованих комплексів розміщено у відкритому репозиторії GitHub за адресою: <https://github.com/entaytion/ugv-object-detection-system> (дата звернення: 29.05.2026).



Репозиторій містить скрипти навчання та валідації моделей RT-DETR-L і YOLOv8-L, модуль інференсу з евристичним фільтром  $H(p)$ , графічний інтерфейс NRKDetectorApp, скрипти перевірки датасету та формування звітів, а також конфігураційні файли та інструкцію з розгортання у файлі README.md.

## Додаток Б

### Презентаційні матеріали

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень

Виконав:  
студент групи КН-22-2  
О.С. Кулініч

Керівник:  
Доцент кафедри КН  
О.А. Пасічник

## 02 АКТУАЛЬНІСТЬ

**≥20**

типів НРК ворога  
використовується  
в бойових умовах

**32**

моделі НРК іден-  
тифіковано

**>70**

наземних платформ  
кодифіковано для  
ЗСУ

**≈200**

найменувань  
фінансується  
Україною

### Основні виклики:

- › Варіативність ракурсів та камуфляж
- › Схожість до цивільної техніки
- › Динамічне освітлення, дим, рослинність
- › Необхідність реального часу

03

## ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

### ОБ'ЄКТ

Процес ідентифікації наземних роботизованих комплексів на основі аналізу зображень

### ПРЕДМЕТ

Методи та алгоритми детекції об'єктів на зображенні

### Архітектури дослідження

**RT-DETR-L**

Основна модель

**YOLOv8-L**

Baseline

04

## МЕТА ТА ЗАВДАННЯ

### МЕТА

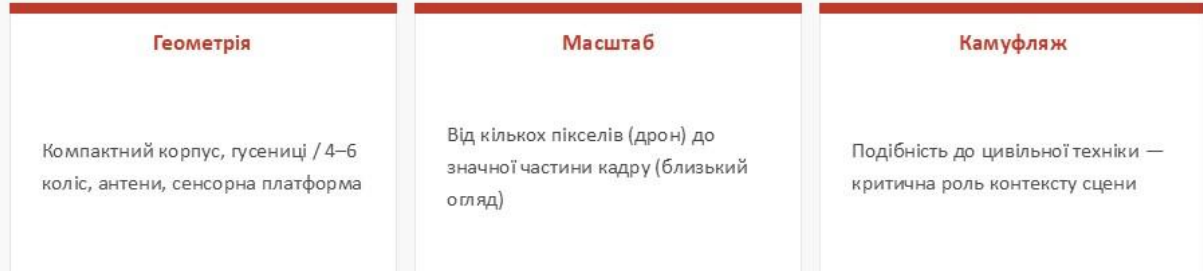
Підвищення точності ідентифікації та швидкості виявлення НРК на зображеннях шляхом розроблення нейромережевого методу.

- 1 Провести аналіз сучасного стану технологій та підходів до ідентифікації наземних роботизованих комплексів за допомогою методів комп'ютерного зору та глибокого навчання
- 2 Формалізувати задачу ідентифікації наземних роботизованих комплексів як задачу детекції об'єктів на зображенні
- 3 Розробити метод ідентифікації наземних роботизованих комплексів на основі аналізу зображень, спрямований на підвищення точності та швидкості виявлення
- 4 Виконати програмну реалізацію методу
- 5 Провести експериментальне дослідження методу з використанням метрик точності та оцінкою швидкості виявлення

## Інформаційна модель ідентифікації НРК



## Специфіка об'єктів класу НРК:



## Загальна схема та алгоритм роботи методу

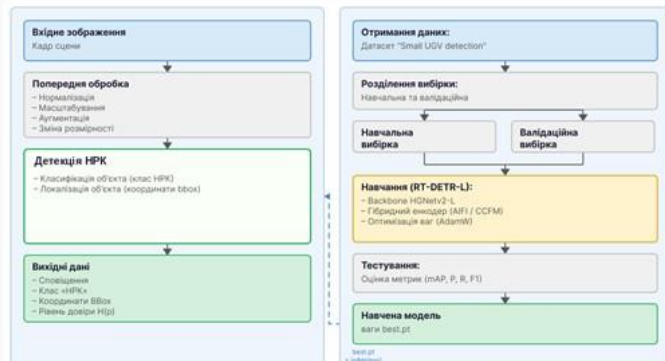


Рисунок 2.1 – Загальна схема нейромережевого методу ідентифікації НРК

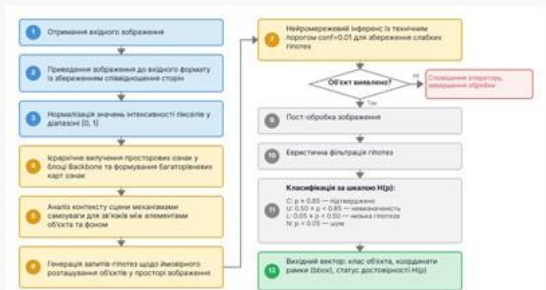


Рисунок 2.2 – Алгоритм роботи методу детекції

07

## ОГЛЯД АРХІТЕКТУР НЕЙРОМЕРЕЖ

Модель	mAP@0.5:0.95	FPS (T4)	Параметри	NMS
Faster R-CNN	37.4	~17	41.8 М	Так
YOLOv5l	49.0	~55	46.5 М	Так
YOLOv8m	50.2	~80	25.9 М	Так
Deformable DETR	46.2	~25	40.1 М	Ні
RT-DETR-X	54.8	~74	67.0 М	Ні
<b>RT-DETR-L ★</b>	<b>53.0</b>	<b>~114</b>	<b>32.0 М</b>	<b>Ні</b>

★ Обрана архітектура: найкраще співвідношення точності, швидкодії та обчислювальних витрат

08

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Трирівнева архітектура NRKDetectorApp

## UI Layer

Рівень взаємодії з користувачем

tkinter · NRKDetectorApp · завантаження зображень · відображення рамок та confidence

## NN Layer

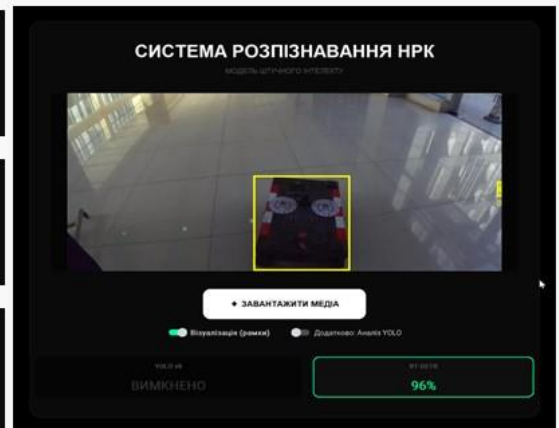
Рівень нейромережевого аналізу

ObjectDetector · RT-DETR-L (основна) · YOLOv8-L (base line) · HeuristicFilter H(p)

## Data Layer

Рівень управління даними

best.pt / last.pt · data.yaml · CSV-звіти · DatasetValidator



Стек технологій:

Python 3.11 · PyTorch 2.5 · Ultralytics 8.4 · OpenCV · NumPy · Pillow

## Архітектура та графічний інтерфейс NRKDetectorApp

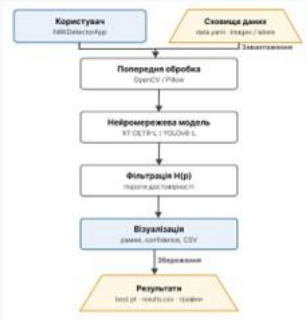


Рисунок 3.1 – Функціональна архітектура програмної реалізації

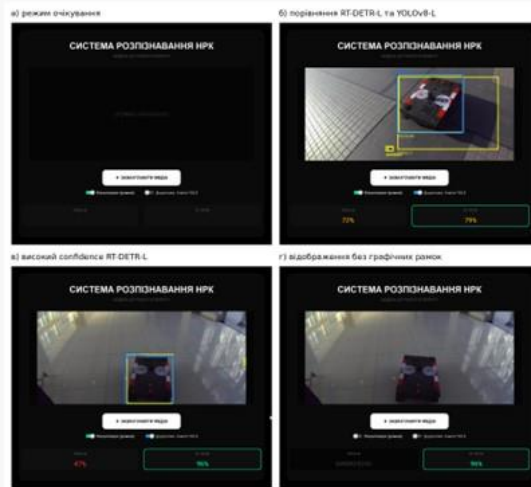
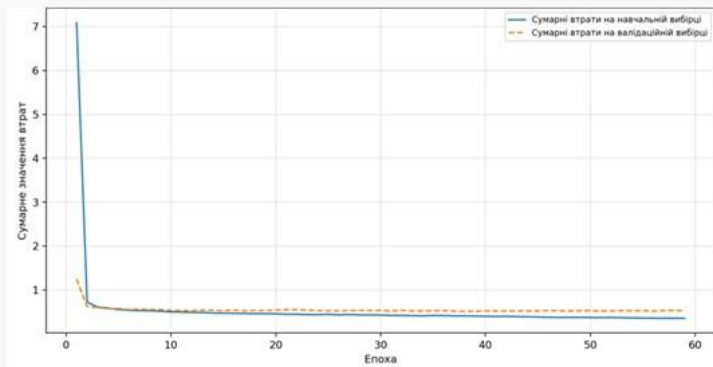


Рисунок 3.9 – Сценарії роботи графічного інтерфейсу застосунку

Таблиця 3.1 — Результати перевірки на валідаційній вибірці

Модель	Precision	Recall	F1-score	mAP@0.5	mAP@0.5:0.95	Інференс, мс
YOLOv8-L	0.981	0.996	0.988	0.991	0.888	20.0
<b>RT-DETR-L</b>	<b>0.989</b>	<b>0.999</b>	<b>0.994</b>	<b>0.992</b>	<b>0.893</b>	<b>17.2</b>



Параметри навчання:

Epochs: 100 (Early Stop)

Batch: 4

Optimizer: AdamW

LR: 0.0001

Warmup: 5 epochs

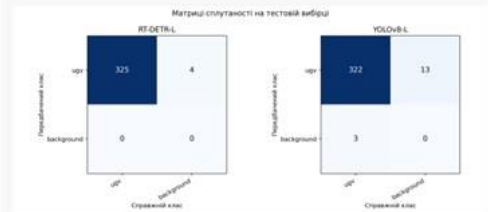
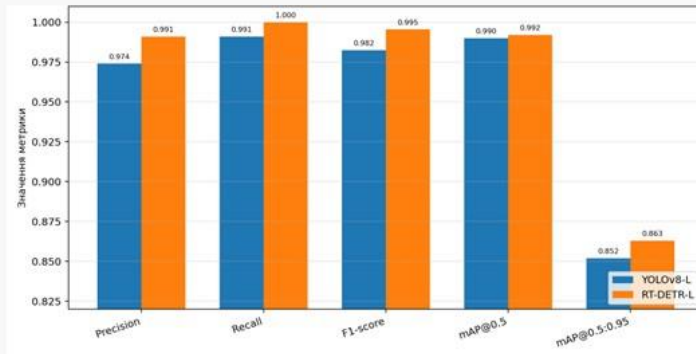
AMP: disabled

RT-DETR-L: зупинено на 59-й епосі (Early Stopping), найкращий чекпоінт — 39-та епоха.

## 11 РЕЗУЛЬТАТИ ТЕСТУВАННЯ МЕТОДУ

Таблиця 3.2 — Порівняльна характеристика на тестовій вибірці (UGV-KPNet, 328 зображень)

Модель	Precision	Recall	F1-score	mAP@0.5	mAP@0.5:0.95	Інференс, мс
YOLOv8-L	0.974	0.991	0.982	0.990	0.852	20.6
<b>RT-DETR-L</b>	<b>0.991</b>	<b>1.000</b>	<b>0.995</b>	<b>0.992</b>	<b>0.863</b>	<b>17.1</b>



## ВИСНОВКИ

- 1 Проаналізовано архітектури детекторів. RT-DETR-L обрана як оптимальна: 32 М параметрів, ~114 FPS, без NMS-постпроцесингу.
- 2 Формалізовано задачу ідентифікації НРК як однокласову детекцію об'єкта ugv з тривірневою шкалою достовірності  $H(p)$ .
- 3 Розроблено неймережевий метод та реалізовано прототипний застосунок NRKDetectorApp з графічним інтерфейсом.
- 4 Тестування на UGV-KPNet (328 зображень): Precision=0.991, Recall=1.000, mAP@0.5:0.95=0.863 за 17.1 мс.
- 5 RT-DETR-L перевершила YOLOv8-L за всіма метриками, не допустила жодного пропуску (0 FN).



Thu Jun 04 18:09:12 EEST 2026, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (http://ap.km.ua) v-16.718

**Максимальне співпадіння з одним документом 2.0%**

Словники перевірки: UA, US, RU. Помилки в документах: 16%

ID: 273657 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень Додано в БД: 2026-06-04 Автора: Олексій КУЛІНІЧ Керівники: Олександр ПАСІЧНИК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	83800	702	2937 (4%)	43 (6%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Олексій КУЛІНІЧ

**Співавтор:**

**Назва:** КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень

**Науковий керівник:** Олександр ПАСТІЧНИК, к.т.н., доцент

**Підрозділ:** Кафедра комп'ютерних наук

**Коефіцієнт подібності 1:** 4.62%

**Коефіцієнт подібності 2:** 2.16%

**Мікропробіли:** 0

**Заміна букв:** 10

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-06-05 01:50:38.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-06-05

Дата

*Петравський Р.Р.*

експерт

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень

Автор: Кулініч Олексій Сергійович

Освітня програма: Комп'ютерні науки

Рівень вищої освіти: Перший (бакалаврський)

Спеціальність: 122 - Комп'ютерні науки

Науковий керівник: к.т.н., доцент Пасічник Олександр Анатолійович

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованими програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються деталі та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

Запозичення, виявлені в роботі Олексія Кулініча, не є плагіатом, оскільки запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту:

Обсяг запозичень, визначений системами виявлення збігів/ідентичності:

- за системою Anti-Plagiarism: 2,0 %;

- за системою StrikePlagiarism КП1: 4,62%; КП2: 2,16%.

Завідувач кафедри КН

Гарант освітньої програми

Керівник кваліфікаційної роботи

Олександр БАРМАК

Олександр МАЗУРЕЦЬ

Олександр ПАСІЧНИК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МОН УКРАЇНИ



Кафедра комп'ютерних наук

## ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента *гр. КН-22-2 Кулініча Олексія Сергійовича*

за темою *Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень*

### 1. Актуальність теми

*Глибоке навчання та комп'ютерний зір критично важливі для автоматизації ситуаційної обізнаності та навігації в умовах високих ризиків. Створення адаптованого нейромережевого методу ідентифікації наземних роботизованих комплексів (НРК) є актуальним науково-практичним завданням. Воно ускладнюється факторами камуфляжу, ракурсів і завад, де традиційні алгоритми (HOG, SIFT) є неефективними. Актуальність роботи підтверджується масштабним впровадженням робототехніки у російсько-українській війні. Ворог використовує понад 30 моделей НРК, тоді як Україна (через кластер Brave1) кодифікувала понад 70 платформ. Це трансформувало бойові дії та вимагає систем детекції з мінімальною затримкою.*

### 2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

*За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є підвищення точності ідентифікації та швидкості виявлення наземних роботизованих комплексів на зображеннях шляхом розроблення нейромережевого методу. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробках у галузі комп'ютерних наук. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.*

### 3. Професійні та особистісні якості бакалавра

*При роботі над кваліфікаційною роботою бакалавра Олексій Кулініч проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при*

розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував професійні компетентності за напрямком «Комп'ютерні науки».

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

**5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

**6. Повнота та якість розкриття теми роботи**

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та виконано програмну реалізацію для валідації та тестування запропонованого метода.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

**8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений у роботі метод та його програмна реалізація можуть бути використані Силами Оборони України для підвищення ефективності виявлення наземних роботизованих комплексів ворога.

**9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



канд. техн. наук, доцент каф. КН Олександр ПАСІЧНИК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МОН УКРАЇНИ

Кафедра комп'ютерних наук



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента гр. КН-22-2 Кулініча Олексія Сергійовича

за темою: Метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень

#### 1. Актуальність обраної теми

Станом на теперішній етап російсько-української війни у бойових умовах зафіксовано застосування противником щонайменше 20 типів наземних роботизованих комплексів, а для потреб Сил Оборони України кодифіковано понад 70 наземних роботизованих платформ понад 10 основних класів, що свідчить про поступову трансформацію характеру ведення бойових дій у напрямі роботизованих систем. У цих умовах ефективність протидії значною мірою визначається можливістю оперативного та високоточного виявлення ворожих НРК з подальшим формуванням рішень у реальному масштабі часу. Інтенсивний розвиток методів аналізу зображень суттєво підвищує рівень автоматизації задач аналізу та обробки візуальної інформації.

#### 2. Повнота розкриття мети та завдань роботи

Мета роботи розкрита повністю, всі завдання виконані.

#### 3. Зміст кожного розділу роботи

В першому розділі виконано аналіз сучасних підходів до вирішення завдань предметної області та сучасного стан. Визначено мету роботи та виконано постановку завдань. В другому розділі сформульовано метод нейромережевої ідентифікації наземних роботизованих комплексів за аналізом зображень. Визначено критерії оцінки точності. В третьому розділі виконано експериментальне тестування методу та виконано оцінку точності ідентифікації.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений у роботі метод та його програмна реалізація можуть бути використані Силами Оборони України для підвищення ефективності виявлення наземних роботизованих комплексів ворога.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

#### 6. Недоліки кваліфікаційної роботи бакалавра

При реалізації методу використано лише дві моделі нейронних мереж.

#### 7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на

яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

Олег ПАВЛОВ, к.т.н., доц, доцент ТМІТ, ХНУ