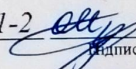
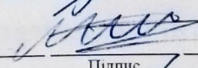
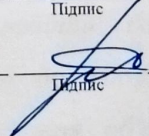


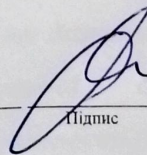
## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

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

Виконала: студент групи КН-21-2  Іван ОЛІЙНИК  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ  
Керівник: д.т.н., проф. каф. КН  Едуард МАНЗЮК  
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

17 06 2025 р.

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  
« 10 » 02 2025 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт»

2. Завдання видано студенту Івану Олійнику  
(Ім'я, прізвище)

3. Керівник роботи д.т.н., проф. каф. КН Едуард МАНЗЮК  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від «07» 02 2025 р. № 23

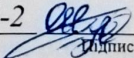
5. Дата видачі завдання студенту: «10» 02 2025 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

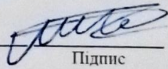
*Завданням кваліфікаційної роботи бакалавра є: обґрунтування актуальності теми, аналіз підходів до побудови методів розпізнавання об'єктів, розробка методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, розроблення функціональних модулів які дозволяють реалізувати даний метод, проведення тестування програмної системи, та проведення експериментальних досліджень ефективності розробленого методу.*

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2025	
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	
3	Проектування та розробка загальної архітектури методу, інформаційної системи та програмного середовища реалізації	березень 2025	
4	Створення та тестування програмного забезпечення	квітень 2025	
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно вимог	травень 2025	
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	
8	Підготовка до захисту та захист кваліфікаційної роботи бакалавра	червень 2025	

Виконавець: студент групи КН-21-2   
Група виконавця

Іван ОЛІЙНИК  
Ім'я, ПРІЗВИЩЕ

Керівник: д.т.н., проф. каф. КН   
Науковий ступінь, посада

Едуард МАНЗЮК  
Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-21-2  
Іван ОЛІЙНИК

Керівник кваліфікаційної роботи бакалавра: д.т.н., проф. каф. КН Едуард  
МАНЗЮК

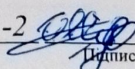
Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
58	23	0	42	2

Мета кваліфікаційної роботи бакалавра – підвищення ефективності пошукових робіт і розслідувань шляхом розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів. Для реалізації методу використано архітектуру YOLOv8, фреймворк PyTorch, бібліотеки комп'ютерного зору та інструменти візуалізації.

Розроблений метод забезпечує високу точність виявлення об'єктів навіть за умов низької якості відео, зашумленого фону або складного освітлення. Завдяки ефективному використанню сучасних нейронних мереж він демонструє мінімальну затримку під час обробки відеопотоку, що є критично важливим для застосування в оперативних сценаріях, зокрема пошуково-рятувальних або розслідувальних операціях.

Ключові слова: комп'ютерний зір, розпізнавання об'єктів, YOLOv8, БПЛА, реальний час, PyTorch.

Виконавець: студент групи КН-21-2   
Група виконавця

Іван ОЛІЙНИК  
Ім'я, ПРІЗВИЩЕ

## Зміст

Вступ.....	4
Розділ 1 Аналіз методів і засобів розпізнавання об'єктів у відеопотоці БПЛА ....	5
1.1 Сучасні підходи до розпізнавання об'єктів комп'ютерного зору БПЛА.....	5
1.2 Теоретичний підхід до обробки відеоданих БПЛА для розпізнавання об'єктів у реальному часі .....	8
1.3 Аналіз існуючого програмного забезпечення для розпізнавання об'єктів у відеопотоці .....	10
1.4 Мета, задачі та вимоги до розробки методу.....	13
Розділ 2 Проектування методу.....	14
2.1. Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт.....	14
2.2 Функціональна структура методу .....	17
2.3 Розробка архітектури нейронної мережі .....	20
2.4 Підготовка робочих вхідних даних для системи .....	28
2.5 Особливості використання спеціалізованих програмних компонентів .....	31
2.6 Висновки до розділу 2 .....	32
Розділ 3 Експериментальне дослідження методу та його програмна реалізація	33
3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення .....	33
3.2 Вибір засобів розробки методу.....	34
3.3 Структура та функціональне призначення програмних складових системи	37
3.4 Особливості реалізації програмних складових системи.....	38
3.5 Аналіз функціональності системи.....	45
3.6 Результати досліджень .....	46
3.7 Висновки до розділу 3 .....	52
Висновок .....	54
Перелік посилань.....	55
Додатки	

### Перелік скорочень

Скорочення, термін, позначення	Пояснення
БПЛА	Безпілотний літальний апарат
ПК	Персональний комп'ютер
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector
R-CNN	Region-based Convolutional Neural Network
MSS	Multi Screen Shot
OpenCV	Open Source Computer Vision Library
VRX	Video Receiver
VTX	Video Transmitter
MFNet	Multi-Feature Network
EDNet	Edge-Optimized Detection Network
AI	Artificial Intelligence
VC	Version Control
ONNX	Open Neural Network Exchange
ОС	Операційна система
NMS	Non-Maximum Suppression
SPPF	Spatial Pyramid Pooling – Fast
COCO	Common Objects in Context

## Вступ

Зростання ролі БПЛА у цивільній, оборонній та рятувальній сферах зумовлює потребу в ефективному виявленні та розпізнаванні об'єктів у реальному часі. Самі БПЛА виступають як засоби збору відеоданих, що потребують оперативної обробки з мінімальною участю людини.

Саме через потребу своєчасного отримання інформації про людей, транспорт, інші БПЛА чи споруди в кадрі тема є актуальною. Проблема полягає в реалізації високоточної системи розпізнавання в умовах обмежених ресурсів БПЛА. Перспективним є підхід, в випадках коли відеопотік із БПЛА передається на ПК для обробки за допомогою глибоких нейронних мереж родини YOLO. Це забезпечує високу точність без зміни апаратної частини БПЛА, що критично для моніторингу, розслідувань і рятувальних операцій.

**Об'єкт дослідження** – процес розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів.

**Предмет дослідження** – методи та технології комп'ютерного зору для розпізнавання об'єктів у завданнях розслідувань і пошукових робіт із використанням безпілотних літальних апаратів.

**Мета кваліфікаційної роботи бакалавра** – підвищення ефективності пошукових робіт і розслідувань шляхом розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів.

**Завдання кваліфікаційної роботи бакалавра** – провести аналіз актуальності з завдання розробки методу, навести дослідження що до наявних підходів що до предметної області цього методу, розробити метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, розробити інформаційну систему яка дозволяє реалізувати даний метод, провести тестування програмної системи, провести експериментальні дослідження ефективності розробленого методу

## **Розділ 1 Аналіз методів і засобів розпізнавання об'єктів у відеопотоці БПЛА**

### **1.1 Сучасні підходи до розпізнавання об'єктів комп'ютерного зору БПЛА**

Стрімкий розвиток із широким застосуванням БПЛА в цивільній, оборонній та промисловій сферах зумовлює потребу в інтелектуальних системах комп'ютерного зору для автоматизованої обробки візуальної інформації з бортових сенсорів. Предметна область охоплює використання алгоритмів глибокого навчання для розпізнавання об'єктів у реальному часі, що в свою сергу дедалі активніше впроваджується в різних галузях.

Серед прикладів практичного використання можна відзначити успішні операції з пошукових робіт загублених осіб у гірських районах Карпат із використанням дронів, оснащених моделями YOLOv5 і тепловізорами камерами [1]. У 2023 році рятувальні служби у США впровадили систему аероконтролю пожеж, де БПЛА автоматично розпізнають осередки займання, та фіксують координати з подальшим передаванням даних в аналітичну систему, що значно зменшило час реагування [2]. У аграрній сфері в Іспанії безпілотники з модулями розпізнавання захворювань винограду за станом листя дозволили знизити використання пестицидів на 18% протягом одного сезону [3].

Окрему увагу заслуговує впровадження подібних систем в оборонному комплексі, зокрема під час спостереження за пересуванням техніки, ідентифікації потенційно небезпечних об'єктів. Алгоритми детекції адаптовані до використання в умовах зниженої видимості або поганої якості відео, але не зважаючи на перешкоди, вони демонструють високу ефективність у польових умовах. У 2024 році було зафіксовано успішне застосування системи з YOLOv8 на базі NVIDIA Jetson для автоматичного визначення рухомих цілей у нічний час у зоні бойових дій [4].

За даними аналітичного звіту Європейської комісії, до 2025 року понад 70% усіх аерофотозйомок у Європейському Союзі здійснюватимуться із

застосуванням безпілотних літальних апаратів. Таке зростання популярності БПЛА пояснюється потребою в оперативному зборі інформації в реальному часі під час моніторингу, екстрених ситуацій, охорони навколишнього середовища та розслідувань. Однак масове використання БПЛА супроводжується технічними обмеженнями, зокрема недостатньою обчислювальною потужністю на борту, що ускладнює виконання складних аналітичних завдань під час польоту [5], [6].

Одним із найперспективніших способів подолання цих обмежень є перенесення інтелектуальної обробки даних на наземні станції. Передача відеопотоку через VRX-приймач на ПК дозволяє використовувати ресурсоємні методи глибокого навчання, тим самим не навантажуючи апаратне забезпечення БПЛА. Такий підхід не лише оптимізує витрати енергії, але й відкриває можливості для використання складних нейронних мереж, зокрема архітектур сімейства YOLO [7].

YOLOv8, як одна з найсучасніших архітектур детекції об'єктів відзначається високою ефективністю в реальному часі, що робить її придатною до практичного використання в змінних та непередбачуваних умовах. Її адаптивність до коливань освітлення, зашумленості та зміни фону дозволяє підтримувати стабільну якість розпізнавання. У наукових працях акцентується увага на важливості модульної побудови подібних систем, що забезпечує гнучкість, ізольовану оптимізацію окремих компонентів та спрощує інтеграцію з іншими підсистемами. [7].

Така структурованість сприяє не лише масштабованості, а й підвищенню стабільності програмного забезпечення в умовах змін середовища. Особливо це актуально при розробці систем які повинні працювати в реальному часі не допускаючи значних затримок або втрат кадрів.

Для реалізації подібних рішень зазвичай використовується стек бібліотек на Python. Зокрема, PyTorch застосовується для побудови та навчання моделей глибокого навчання, Ultralytics – для запуску YOLOv8, OpenCV – для базової обробки відеопотоку, MSS – для захоплення зображень з екрана, а PyQt5 – для створення графічного інтерфейсу користувача. Така комбінація дозволяє

реалізувати повноцінну систему обробки зображень, яка працює на стороні оператора та може функціонувати незалежно від обмежень БПЛА [8], [9], [10], [11], [12].

Функціональне середовище експлуатації таких систем зазвичай передбачає участь людини на етапі контролю або прийняття рішень. Залежно від завдань, до процесу можуть бути залучені оператор БПЛА, технічний спеціаліст, аналітик відеоінформації, інженер або співробітник, що займається маркуванням і передачею об'єктів для подальшого аналізу. Усе частіше частина цих процесів автоматизується, тим самим знижуючи когнітивне навантаження на персонал і дозволяючи зосередитися на критичних моментах [13].

Семантична модель системи розпізнавання об'єктів на основі відеопотоку з БПЛА передбачає чітке визначення типів вхідних і вихідних даних, а також логіки їх обробки. Відеопотік трансформується у послідовність кадрів, де в свою чергу кожен з яких подається на обробку як окрема інформаційна одиниця. Кожен кадр може містити кілька виявлених об'єктів, де кожен описується координатами обмежувальної рамки, класом, рівнем впевненості, а також часом фіксації. Такий підхід дозволяє ефективно індексувати та аналізувати події у просторі-часі, що є критично важливим у сценаріях, що представлені як швидкий рухомий або тимчасово зникаючий із поля зору камери об'єкт [14].

У реалізації таких систем ключовими є не лише точність виявлення, але й швидкодія та енергоефективність. Дослідження показують, що оптимізовані моделі YOLO разом із методами зменшення розміру даних та апаратним прискоренням дозволяє обробляти кадри швидше без суттєвого падіння точності. Водночас впровадження адаптивного керування частотою оновлення кадрів і умовної активації моделей підвищує енергоефективність, що є критичним для тривалої автономної роботи. [15].

Таким чином, інтелектуальні системи розпізнавання об'єктів із використанням БПЛА є не лише інструментами високотехнологічного нагляду, а й важливим компонентом забезпечення безпеки, продуктивності та реагування у критичних ситуаціях. Подальші дослідження та розробки зосереджені на

підвищенні швидкодії, точності розпізнавання, а також зменшенні енергоспоживання й автономізації таких платформ.

## **1.2 Теоретичний підхід до обробки відеоданих БПЛА для розпізнавання об'єктів у реальному часі**

У сучасній науково-практичній літературі та інженерній практиці спостерігається активне зростання зацікавленості до застосування комп'ютерного зору в завданнях, що здійснюють обробку відеопотоків із камер спостереження, БПЛА на мобільних платформах. Це зумовлено як розвитком апаратного забезпечення, так і стрімким прогресом у сфері машинного та глибокого навчання. Системи, що інтегрують комп'ютерний зір, мають високий потенціал у різних сферах, як безпеки й оборони до сільського господарства, логістики та екологічного моніторингу.

Одним із ключових напрямів є розробка алгоритмів, які здатні ефективно аналізувати вхідні відеодані в режимі миттєвої обробки. Такі системи мають функціонувати в умовах обмежених обчислювальних ресурсів, змінних умов освітлення, різних ракурсів і шумів. Як зазначено в особливій актуальності ці технології набувають у критичних сценаріях під час пошуково-рятувальних операцій, військових дій, реагування на надзвичайні ситуації та охорони стратегічних об'єктів інфраструктури. У таких випадках важливо не лише виявити об'єкт, а й зробити це з мінімальною затримкою та максимальною точністю [16].

Теоретичною основою побудови таких інтелектуальних систем слугують згорткові нейронні мережі, які зарекомендували себе як надзвичайно ефективні в завданнях класифікації, детекції та семантичної сегментації об'єктів. Вони здатні автоматично виявляти характерні ознаки зображення, формуючи багаторівневу абстракцію, що дозволяє моделі адаптуватися до складних і варіативних сценаріїв реального світу. Згідно з архітектури YOLO, SSD та Faster R-CNN є серед найбільш вживаних і досліджених рішень у сфері комп'ютерного зору, особливо в задачах із пріоритетом на швидкодію [9].

Серед них YOLO виділяється як архітектура, яка дозволяє здійснювати детекцію об'єктів за один прохід нейронною мережею, що забезпечує високу швидкість обробки. Нова версія YOLOv8 повністю описана в офіційній документації Ultralytics як високопродуктивна, гнучка та масштабована модель, що підтримує прискорення на графічних процесорах, має компактний дизайн і придатна для розгортання на мобільних чи вбудованих пристроях. Вона також інтегрує підтримку декількох конфігурацій, це найменша YOLOv8n до найпотужнішої YOLOv8x. Завдяки цьому інженери можуть обирати відповідну модель залежно від цільової платформи [17].

Навіть найменша версія моделі YOLOv8n може демонструвати високу продуктивність при достатньому обладнанні, при цьому забезпечуючи до 45 кадрів/с при точності близько 0.5 mAP, що дозволяє ефективно працювати в умовах обмежених ресурсів як: автономні БПЛА, мобільних роботах чи портативних камерах. Високу якість виявлення забезпечують сучасні підходи до навчання: псевдо-розмітка, data augmentation, а також генерація синтетичних зображень для покриття рідкісних сценаріїв. Це знижує залежність від великих вручну розмічених вибірок і підвищує адаптивність системи. Використання трансферного навчання додатково сприяє ефективному донавчанню моделей. Подібні підходи широко застосовуються, зокрема, у сфері аграрного моніторингу з БПЛА, де важливою є швидкість і надійність обробки зображень [18], [19].

Ефективна реалізація інтелектуальних систем значною мірою залежить від вибору мови програмування та супутніх інструментів. На практиці найчастіше використовують Python завдяки лаконічному синтаксису. Зокрема, OpenCV забезпечує широкий набір алгоритмів для попередньої обробки зображень, таких як фільтрація, зміна кольору, контурна, детекція й трекінг. MSS в свою чергу корисний для захоплення відео під час тестування або збору даних. PyQt5 дозволяє створювати інтерактивні графічні інтерфейси. [20].

У структурному плані переважає модульний підхід до побудови програмної системи. Виділення окремих рівнів при захопленні відео, попередня обробка, детекція об'єктів, візуалізація результатів. Це надзвичайно важливо для

забезпечення масштабованості, адаптивності та надійності, особливо в умовах розгортання на різних апаратних конфігураціях.

Таким чином, у контексті подальшої розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, доцільним буде використання зв'язки з YOLOv8n як основної моделі детекції, також Python як основної мови програмування з підтримкою численних бібліотек. Такий підхід забезпечує прийнятний компроміс між точністю, швидкістю виконання та простотою впровадження, що робить його ефективним вибором для розробки мобільних інтелектуальних систем миттєвої обробки.

### **1.3 Аналіз існуючого програмного забезпечення для розпізнавання об'єктів у відеопотоці**

У контексті стрімкого розвитку БПЛА зростає потреба в системах комп'ютерного зору, які здатні обробляти відеопотік у реальному часі для вирішення задач пошуково-рятувальних операцій. Ефективність таких рішень залежить від програмного забезпечення, що має забезпечувати точність, швидкодію, адаптивність до зовнішніх умов і придатність до використання в обмежених апаратних середовищах. Існуючі програмні продукти, які охоплюють широкий спектр функціоналу, що дозволяє обирати рішення відповідно до конкретних потреб.

GeoDeer – це платформа для обробки зображень з БПЛА, що інтегрується з WebODM і використовує моделі YOLOv7 та DeepForest RetinaNet [21]. Вона працює зі статичними знімками, а не з відео в реальному часі, що обмежує її ефективність у сценаріях, де потрібна миттєва реакція (рисунок. 1.1). Відсутність підтримки стрімінгу та live-аналітики ускладнює використання GeoDeer для моніторингу та оперативного реагування [22].



Рисунок 1.1 – Робота застосунку [22].

SlimYOLOv3 – це модифікація оригінальної YOLOv3 оптимізованої для роботи на БПЛА (рисунок 1.2). Завдяки використанню технік проріджування каналів, модель досягає значного зменшення кількості параметрів та обчислювальної складності без суттєвої втрати точності. Зокрема, SlimYOLOv3 демонструє зменшення FLOPs на 90,8% та кількості параметрів на 92%, працюючи приблизно вдвічі швидше за оригінальну YOLOv3, що робить її придатною для реального часу на обмежених пристроях [23].



Рисунок 1.2. – Робота SlimYOLOv3 [24].

Модель MFNet реалізує підхід мультимодального навчання для виявлення об'єктів у складних умовах. Вона використовує слабко контрольоване навчання з псевдо-мітками, що дозволяє ефективно працювати з обмеженими анотаціями (рисунок 1.3). Система поєднує RGB, теплові та глибинні дані для підвищення точності в умовах низької освітленості або диму. Однак складність реалізації та обчислювальні вимоги обмежують її застосування в edge-середовищах [25].



Рисунок 1.3. – Робота моделі MFNet [26].

Також варто згадати інші актуальні інструменти:

EDNet – легка модель для edge-пристроїв, оптимізована під швидкодію з мінімальними ресурсами. Забезпечує обробку відео з низькою затримкою і точністю mAP  $\sim 0.65$ , ідеальна для автономних БПЛА [27];

FastDet – надшвидка модель для ARM-платформ, базується на MobileNetV3. Працює зі швидкістю до 80 FPS на мобільних пристроях і добре підходить для мобільних додатків і БПЛА із обмеженим залізом [28];

Detectron2 – гнучкий фреймворк від Meta AI для сегментації, детекції та трекінгу. Підтримує найсучасніші моделі, зручний для наукових та практичних застосувань на БПЛА [29];

Roboflow – онлайн-сервіс для збору, обробки й тренування моделей, з підтримкою експорту на edge-пристрої. Простий у використанні, підходить для швидкого розгортання CV-систем [30];

Ultralytics HUB – офіційна платформа YOLOv5/v8 з підтримкою тренування, хмарного запуску та експорту моделей. Має зручний API та інтеграцію з PyTorch, ONNX, TensorRT [31];

Аналіз наявних рішень для розпізнавання об'єктів у відеопотоці з БПЛА показує, що жодне з них наразі не поєднує всі критично важливі характеристики: високу точність, низьку затримку, відкритість, автономність та простоту адаптації. Саме тому розробка нової платформи з використанням YOLOv8n, PyTorch, PyQt5 і модульної архітектури є актуальним напрямком. Така система дозволить поєднати переваги сучасних технологій комп'ютерного зору з вимогами польових застосувань у реальному часі.

## 1.4 Мета, задачі та вимоги до розробки методу

Метою кваліфікаційної роботи бакалавра є підвищення ефективності пошукових робіт і розслідувань шляхом розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів.

Завданням кваліфікаційної роботи бакалавра є:

- провести аналіз актуальності з завдання розробки методу;
- здійснити огляд досліджень існуючих підходів у предметній області методу;
- розробити метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт;
- розробити інформаційну систему яка дозволяє реалізувати даний метод;
- провести тестування програмної системи;
- провести експериментальні дослідження ефективності розробленого методу.

Для реалізації методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт було обрано архітектуру YOLOv8n як основну модель детекції. Мову програмування Python застосовано як базову платформу завдяки її широкій підтримці бібліотек для машинного навчання, комп'ютерного зору та обробки відео. Перед початком розробки було виконано встановлення необхідних бібліотеки, зокрема torch, torchvision, torchaudio, ultralytics, opencv-python, numpy, scipy, matplotlib, pillow, mss, PyQt5, pyqtgraph, pywin32, pygetwindow і pynput, це все були встановлені через командний рядок cmd у попередньо створене віртуальне середовище. Встановлення PyTorch здійснювалось із використанням офіційного CUDA-репозиторію для забезпечення апаратного прискорення обчислень. Така конфігурація гарантує ефективну й стабільну роботу методу в умовах реального часу.

## Розділ 2 Проектування методу

### 2.1. Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

Спостерігаючи за стрімким розвитком сучасних технологій та методів аналізу даних, важливо також розуміти як відеопотік потрапляє до обчислювальної частини системи де в свою чергу застосовуються описані теоретичні підходи функціонування методу. На рисунку нижче наведено узагальнену схему передачі зображення від об'єкта в полі зору БПЛА до комп'ютера, який виконує інтелектуальну обробку відео в реальному часі (рисунок 2.1).

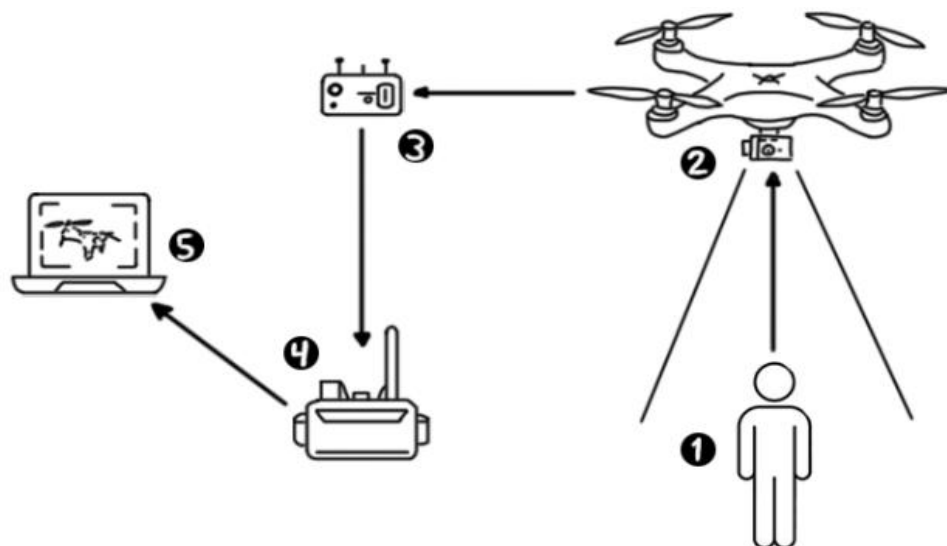


Рисунок 2.1. – Послідовність передачі відеоданих від БПЛА до комп'ютера для обробки в системі комп'ютерного зору.

Схема охоплює п'ять ключових етапів, а саме:

- зображення фіксується камерою БПЛА з високою роздільною здатністю та широким кутом огляду;
- на борту виконується попередня обробка з відеостабілізацією за допомогою одноплатного комп'ютера;

- стабілізоване відео передається через бездротовий VTX-модуль з мінімальною затримкою;
- сигнал приймається наземним VRX-модулем або FPV-окулярами;
- відео надходить на комп'ютер з апаратним декодуванням і буферизацією, після чого обробляється методом.

Данні що надходять на ПК проходять обробку схемою нижче (рисунок 2.2).

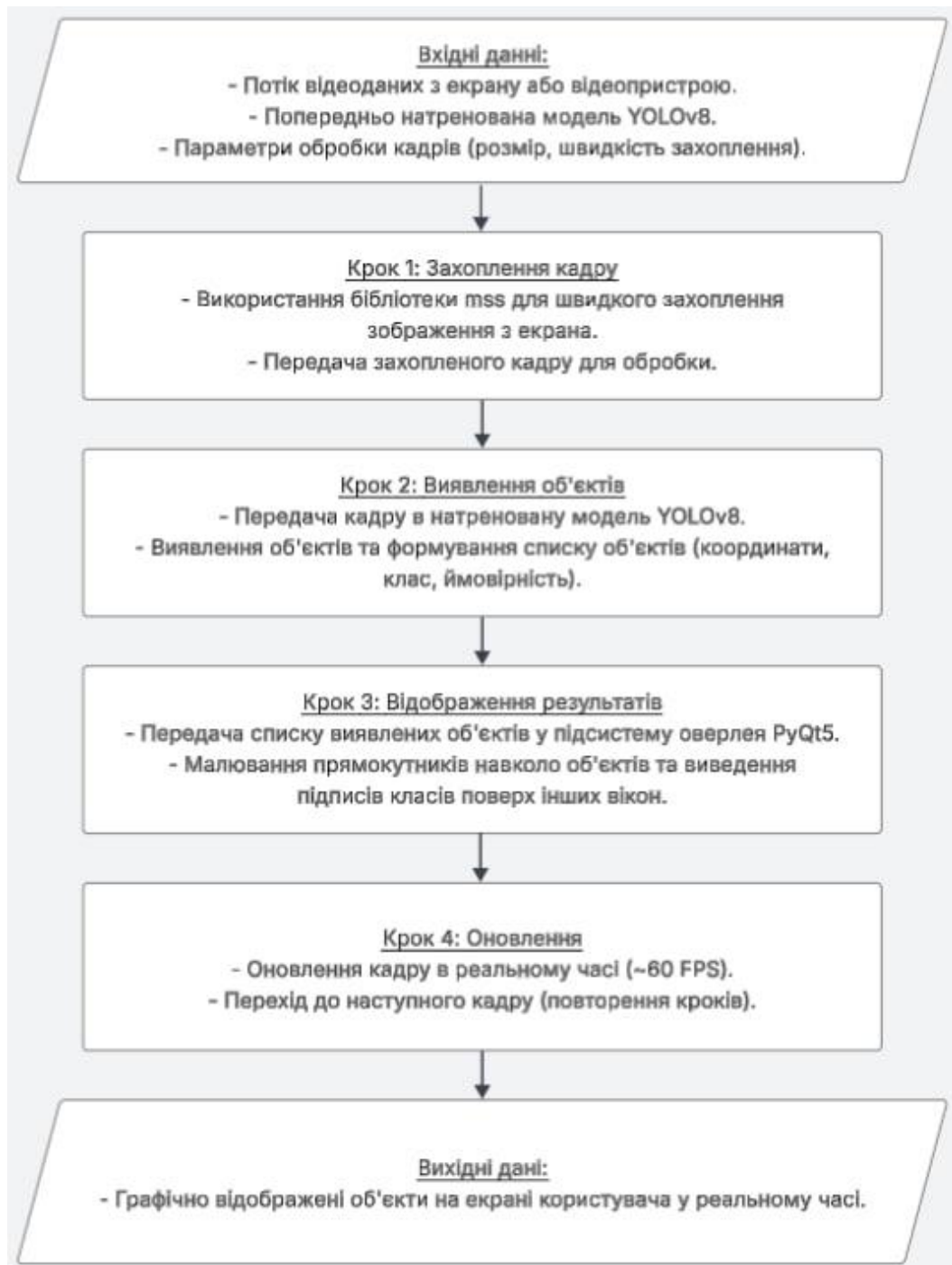


Рисунок 2.2 – Схема методу виявлення об'єктів на відеопотоці.

Схема методу виявлення об'єктів на відеопотоці має наступну архітектуру:

- на вхід системи подається потік відеоданих, що отримані із камери БПЛА. Також використовується попередньо натренована модель YOLOv8 для розпізнавання об'єктів і відповідно налаштовуються параметри обробки кадрів, такі як розмір кадру та частота захоплення зображень;

- на першому кроці здійснюється захоплення зображення з відеопотоку. Для цього використовується бібліотека mss, що забезпечує швидке отримання скріншотів без значних затримок. Захоплений кадр передається на подальшу обробку для виявлення об'єктів;

- після захоплення кадру у третьому кроці, він передається до попередньо натренованої моделі YOLOv8. Модель проводить аналіз зображення і виявляє об'єкти, що присутні на кадрі. Для кожного знайденого об'єкта формується відповідний список параметрів: координати обмежувального прямокутника, клас об'єкта та ймовірність передбачення;

- на четвертому етапі отримані дані про виявлені об'єкти передаються до підсистеми візуалізації, що побудована на основі бібліотеки PyQt5. Система накладає графічні прямокутники навколо виявлених об'єктів і виводить підписи з назвами класів безпосередньо поверх інших елементів інтерфейсу або відеопотоку;

- після відображення результатів кадр оновлюється у реальному часі із високою частотою. Система переходить до захоплення наступного кадру і повторює всі кроки, забезпечуючи безперервне функціонування;

- у підсумку користувач отримує графічно відображені об'єкти на екрані в реальному часі, що дозволяє оперативно приймати рішення під час пошуково-рятувальних або розслідувальних операцій за допомогою безпілотного літального апарата.

Таким чином послідовності передачі відеоданих від БПЛА до комп'ютера для обробки в системі комп'ютерного зору демонструє шлях між фізичним середовищем і програмними модулями на яких зосереджено основну частину розробки. Вона дозволяє краще представити звідки надходять дані, до яких в свою

чергу буде застосовуються схема методу виявлення об'єктів на відеопотоці, яка представлена вище і спроектована таким чином, щоб максимально ефективно обробляти відеопотік та відображати виявлені об'єкти з мінімальними затримками, що є критично важливим для завдань розслідувань і пошукових операцій з використанням БПЛА.

## 2.2 Функціональна структура методу

Проектування ефективної інтелектуальної системи виявлення об'єктів у реальному часі потребує не лише вибору відповідних алгоритмів та моделей, але й створення чіткої модульної інформаційної структури. Така структура має забезпечити максимальну швидкість обробки, гнучкість, стійкість до помилок, можливість масштабування та сумісність з апаратним забезпеченням, зокрема з джерелами відеопотоку з безпілотних літальних апаратів.

Робота системи виявлення об'єктів у реальному часі ґрунтується на узгодженій взаємодії кількох незалежних оброблювальних етапів які послідовно трансформують сирий відеосигнал у зрозумілу, структуровану інформацію для користувача. Кожен етап має чітко окреслене функціональне призначення, від первинного захоплення кадру до графічного представлення виявлених об'єктів на екрані. Такий підхід дозволяє досягти високої ефективності, оскільки спрощує масштабування, тестування та заміну окремих компонентів без потреби модифікувати всю систему. Подібна логіка побудови є типовою для сучасних інтелектуальних додатків, орієнтованих на обробку потоку даних у реальному часі.

У межах кваліфікаційної роботи бакалавра було розроблено функціональні модулі, які орієнтована на роботу з відеопотоком у реальному часі з використанням моделі YOLOv8, бібліотеки захоплення кадрів MSS та технології оверлейної візуалізації на основі PyQt5. Метод передбачає обробку потокового відео на локальному ПК. Архітектура системи представлена на рисунку нижче і

включає шість ключових модулів, взаємодія яких забезпечує повний цикл від захоплення кадру до виведення результатів користувачу (рисунок 2.3).

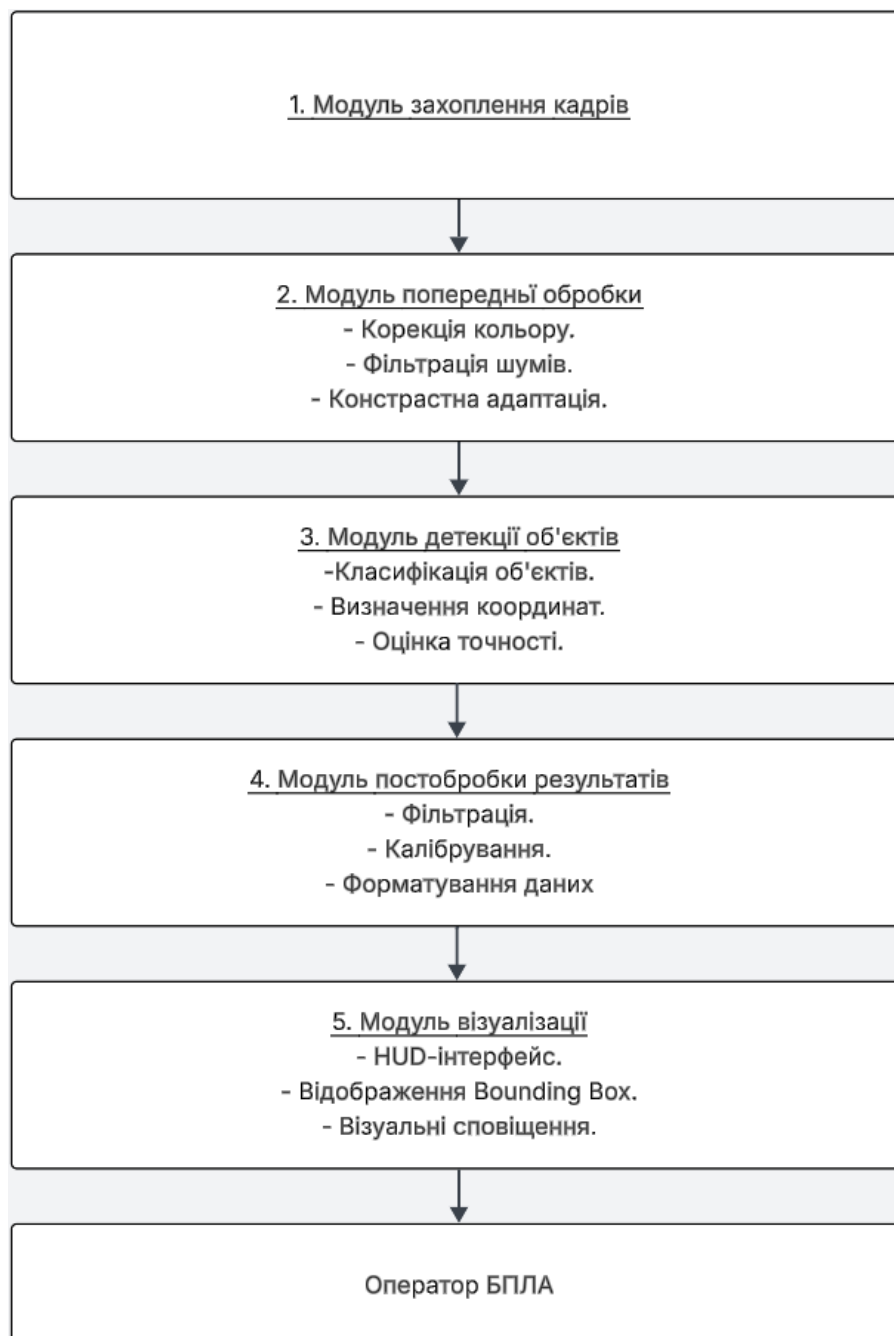


Рисунок 2.3 – Функціональні модулі обробки відеоданих БПЛА.

Покроковий опис функціональних модулів обробки відеоданих БПЛА для методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт:

– перший модуль відповідає за безперервне захоплення зображення з відеопотоку. Застосовується бібліотека MSS, яка дозволяє використовувати API DirectShow, забезпечуючи мінімальну затримку і підтримку високих роздільностей. Головним функціоналом являється: кадрування вікна або екрану; буферизація для уникнення пропуску кадрів; передача кадру в модуль попередньої обробки;

– модулю попередньої обробки перш ніж передати зображення на аналіз для аналізу кадру приходиться пройти базову обробку для покращення якості: корекція кольору: наприклад, використання histogram equalization для покращення освітлених ділянок; фільтрація шумів: зазвичай через Gaussian Blur для зменшення артефактів зображення; підвищення контрасту: покращення виявлення об'єктів на тлі, особливо в складних умовах;

– модуль детекції об'єктів являє собою мозок, у якому виконується основне розпізнавання. Застосовується модель YOLOv8-nano, яка оптимізована для миттєвої обробки з продуктивністю 40–45 FPS на середньому GPU. Головними задачами є: класифікація кожного об'єкта; визначення координат bounding box; розрахунок імовірності (confidence score) для кожного класу; побудова списку об'єктів у форматі: [(x1, y1, x2, y2), class\_id, confidence];

– модуль постобробки результатів може генерувати накладання або повторювані об'єкти, які слід обробити: NMS-фільтрація виконує усунення дублювань; калібрування координат у разі масштабування; форматування даних для передачі у візуалізацію або збереження;

– модуль візуалізації відповідає за відображення результатів поверх відеопотоку, або активних вікон. Реалізується через прозору GUI-обгортку, яка дозволяє показувати: графічні рамки довкола об'єктів; текстові підписи класів; іконки, індикатори, рор-уп повідомлення; інтерактивні елементи управління;

– оператор БПЛА представлений у цій схемі як точка взаємодії. Інформація оператору передається у максимально зручному форматі: голосові або візуальні сповіщення про виявлені об'єкти; виведення метаданих; елементи керування.

Запропоновані функціональні модулі методу базуються на принципах модульності, асинхронності та масштабованості. Кожен модуль відповідає за чітко визначену частину обробки, що дозволяє легко відстежувати помилки, змінювати окремі компоненти, та масштабувати систему під більші навантаження.

Завдяки цій архітектурі досягається низька загальна затримка, стабільність роботи при високому FPS, а також можливість інтеграції з різними типами БПЛА і при цьому не змінюючи логіку ядра системи. Така структура забезпечує високу ефективність і готовність до адаптації під практичні задачі розпізнавання в умовах динамічного середовища.

### **2.3 Розробка архітектури нейронної мережі**

Так як для розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт використовується YOLOv8, це власне як раніше оглянуто було, є одна з найсучасніших архітектур для задач детекції об'єктів у реальному часі. Ця модель є продовженням лінійки YOLO яка отримала широке визнання за баланс між точністю та швидкістю, це особливо помітно у сценаріях де необхідна мінімальна затримка обробки, наприклад таких як відеопотоки з безпілотників. YOLOv8 реалізується у вигляді згорткової нейронної мережі адаптованої до ресурсів середньої потужності, та демонструє відмінні результати на задачах класифікації й локалізації в кадрі (рисунок 2.4).

Модель складається з 24 згорткових шарів та двох повнозв'язних, організованих у послідовну структуру, що перетворює зображення в простір ознак та виконує класифікацію й регресію координат виявлених об'єктів [32].

На вході моделі подається зображення розміром 448x448 пікселів яке проходить через згорткові шари з ядрами 7x7, 3x3 та зменшується із часом. Кожна згортка витягує локальні патерни які є ознаками об'єктів. Шари з ядрами мінімального розміру виконують зменшення розмірності простору ознак без

втрати важливої інформації, що покращує швидкодію моделі, це в свою чергу є критично важливий параметр для задач миттєвої обробки на обмежених ресурсах.

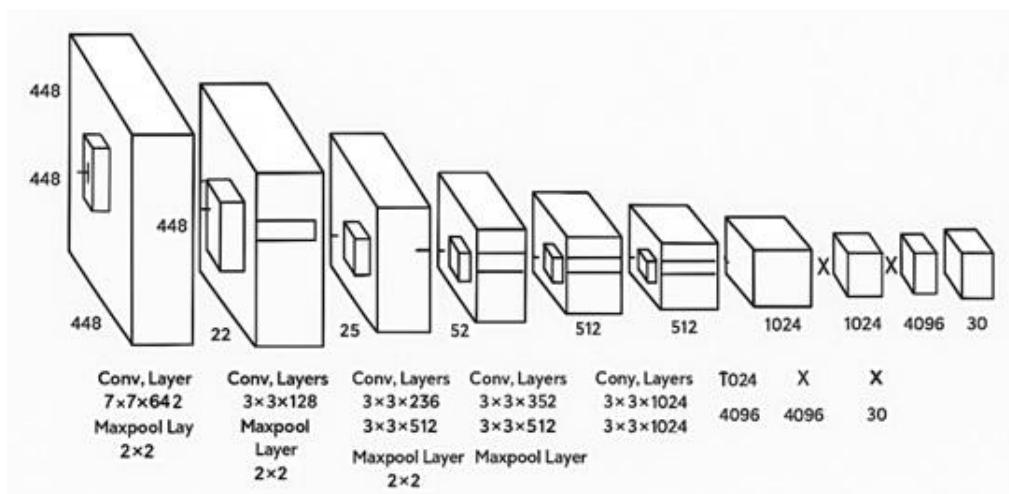


Рисунок 2.4 – Класична архітектура згорткової нейромережі для обробки зображень.

У ключових місцях між згортковими шарами застосовуються операції максимального підвибіркового зниження розміром 2x2. Це дозволяє зменшити просторовий розмір представлення, одночасно зберігаючи семантику об'єктів. Таким чином зменшується кількість параметрів та необхідна оперативна пам'ять для обчислень.

Заключні шари містять повнозв'язні нейрони, це орієнтовно становить один на 4096 елементів та вихідний, що видає координати рамки виявленого об'єкта, клас та ймовірність. Це дозволяє моделі генерувати не тільки клас об'єкта, а й точну локалізацію у кадрі.

Усі шари мережі навчаються за допомогою переднавчання з подальшим донавчанням на спеціалізованих датасетах у форматі YOLO. Такий підхід дає змогу адаптуватися до конкретних умов використання БПЛА [33].

Таким чином, архітектура наведена на рисунку повноцінно ілюструє загальний підхід до побудови детекторів у рамках родини YOLO, а саме поєднання швидкої згорткової обробки, багаторівневого витягання ознак та

регресійного прогнозування просторових координат. Це забезпечує адаптивність системи до широкого спектру умов та потреб польового застосування.

Варто звернути увагу не лише на те як здійснюється розпізнавання, але і на архітектуру YOLOv8 що використовується у методі розпізнавання об'єктів (рисунок 2.5).

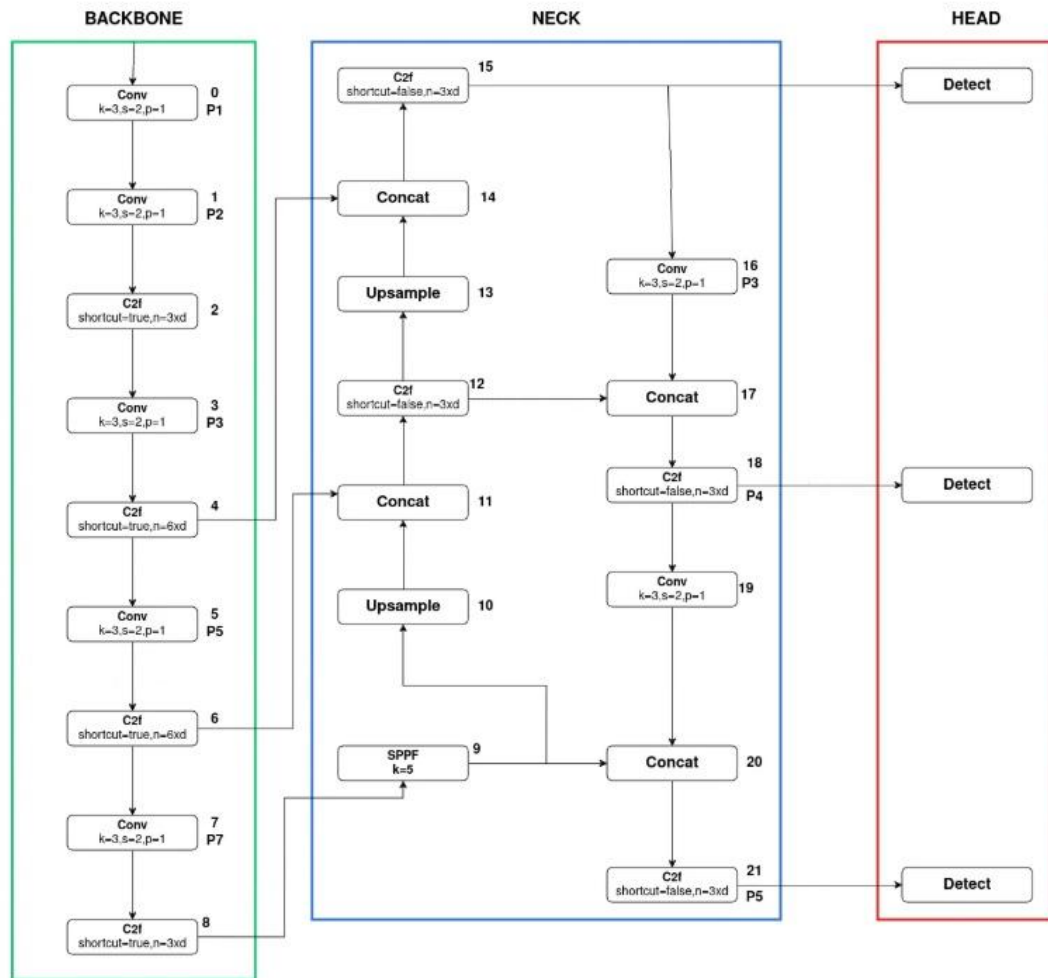


Рисунок 2.5. – Загальна архітектура YOLOv8 та її основні частини.

Архітектура YOLOv8, що використовується у розробленій інформаційній системі розпізнавання об'єктів для безпілотних літальних апаратів побудована на трьох функціонально відокремлених підсистемах: Backbone, Neck та Head. Такий поділ забезпечує модульність, гнучкість і ефективність обробки зображень у реальному часі.

Backbone відіграє роль високоефективного екстрактора ознак із вхідного зображення. Цей компонент відповідає за перетворення сирих піксельних даних у багатовимірні ознаки, що містять структурну, просторову та семантичну інформацію про зображення. В YOLOv8 як базовий блок використовується оптимізований C2f-модуль, який покращує перенесення градієнтів і дозволяє зменшити кількість параметрів без втрати точності.

Neck здійснює агрегацію ознак з різних масштабів, отриманих у Backbone. Вона відповідає за поєднання низькорівневих і високорівневих ознак, що є критично важливим для виявлення об'єктів різного розміру. Зазвичай ця частина архітектури включає підсистеми типу PANet або FPN, однак у YOLOv8 вона реалізована у вигляді покращеної схеми Upsample + Concat з глибокими з'єднаннями.

Head здійснює фінальну обробку ознак та виконує регресію обмежувальних рамок і класифікацію об'єктів. Однією з ключових інновацій YOLOv8 є відмова від традиційної концепції "anchor boxes" та перехід до anchor-free підходу, що дозволяє значно спростити процес навчання та зменшити потребу у вручну налаштованих гіперпараметрах. Детектори на рівнях P3, P4 та P5 відповідають за розпізнавання об'єктів відповідно малого, середнього та великого розміру.

Саме завдяки такій структурі YOLOv8 демонструє високу продуктивність на широкому спектрі задач, включаючи сценарії у режимі миттєвої обробки, що робить її придатною для використання на вбудованих системах, таких як одноплатні комп'ютери NVIDIA Jetson або ноутбуки з GPU. У рамках розробки запропонованої системи використовувалась конфігурація YOLOv8n, що забезпечує баланс між продуктивністю та точністю для задач моніторингу з БПЛА [33].

Подальше поглиблення в архітектуру YOLOv8 дозволяє детально описати особливості її внутрішньої побудови та принципи функціонування, що добре дає зрозуміти блок вище (рисунок 2.6). Вхідним шаром є зображення розміром  $n$  розміру, яке подається до першого згорткового шару, що використовує ядро P1.

Ця операція дозволяє зменшити просторову роздільну здатність удвічі шляхом просторової субдискретизації, забезпечуючи початкове вилучення ознак на базовому рівні.

Кількість каналів у результаті згортки визначається як:

$$\text{out}_{\text{channels}} = \min(64, mc) \cdot w \quad (2.1)$$

де  $mc$  – це максимально допустима кількість каналів, а  $w$  – коефіцієнт ширини, який для моделі YOLOv8n дорівнює 0.25. Таким чином кількість вихідних каналів становить:

$$\min(64, 1024) \cdot 0.25 = 64 * 0.25 = 16 \quad (2.2)$$

У другому функціональному блоці архітектури використовується модифікований модуль, який включає логічний параметр `shortcut` і параметр `n`, що визначає кількість вузьких блоків.

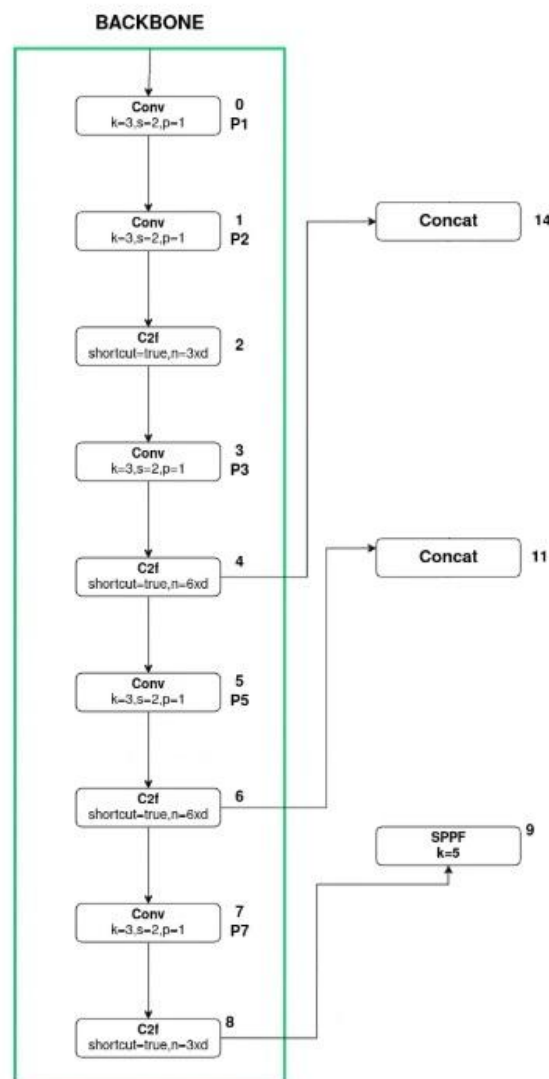


Рисунок 2.6. – Блок YOLOv8, Backbone.

У реалізації YOLOv8 значення  $n$  визначається як:

$$n = 3 * d \quad (2.3)$$

де  $d$  – коефіцієнт глибини. Для моделі типу  $n$  отримаємо:

$$n = 3 * 0.33 = 0.99 \Rightarrow 1 \text{ bottleneck} \quad (2.4)$$

Це означає, що в блоці буде використано один вузький блок, а роздільна здатність карти ознак при цьому зберігається сталою.

Особливу увагу привертає блок SPPF. Його завдання полягає у генерації масштабостійких ознак з урахуванням локальних і глобальних патернів зображення без зміни його геометрії. Це досягається через застосування кількох MaxPooling-операцій з подальшим об'єднанням результатів, що дозволяє зберегти інформацію навіть при сильному зменшенні роздільної здатності [34].

Архітектура YOLOv8 структуровано поділяється ще на два ключові елементи як Neck та Head, кожен з яких відіграє важливу роль у процесі екстракції ознак та прогнозування (рисунок 2.7).

На наданій схемі neck займає центральну частину архітектури, що позначено синім кольором, і побудована за принципами багатопарової агрегації зворотного зв'язку. В основі neck-структури лежить ідея злиття просторово зменшених глибоких ознак з деталізованими вхідними ознаками через поєднання операцій *upsampling* та *concatenation*. Це дозволяє сформувати такі представлення, які містять як детальні краї об'єктів, так і контекстуальні характеристики. З технічної точки зору, що забезпечить проходження зображення через блок SPPF, який застосовує каскадні згортки з ядрами різних розмірів, зберігаючи розмірність виходу, але збагачуючи інформацію про просторові взаємозв'язки. Це дає змогу зловити об'єкти, які не мають чіткої геометричної структури або розміщені на складному фоні.

Після цього глибокі ознаки проходять через ланцюг операцій, де йому приходиться на кожному рівні використовувати інтерполяцію до вищої роздільності, що в свою чергу супроводжується об'єднанням із відповідними тензорами із менш глибоких рівнів. Таким чином, формується трирівнева піраміда ознак, кожен рівень якої містить як локальні, так і глобальні дескриптори сцени.

Архітектурно це забезпечується блоками типу C2f легковими структурами на базі роздільного проходження частин тензора через скорочений шлях з наступною конкатенацією. C2f є вдосконаленням класичних ResNet-подібних bottle-neck блоків, оскільки дозволяє зменшити надмірне дублювання обчислень та зберігає пропускну здатність інформаційного потоку.

Після того як neck сформувала багатопланові тензори ознак, що поєднують локальні й глобальні характеристики, ці тензори спрямовуються до голови архітектури функціонального модуля, який безпосередньо реалізує процес детекції. У YOLOv8 голова представлена у вигляді трьох незалежних детекційних блоків, де кожен з яких спеціалізується на об'єктах певного масштабу. Це архітектурне рішення втілює принцип multi-scale prediction, який є одним із наріжних каменів ефективності моделей YOLO.

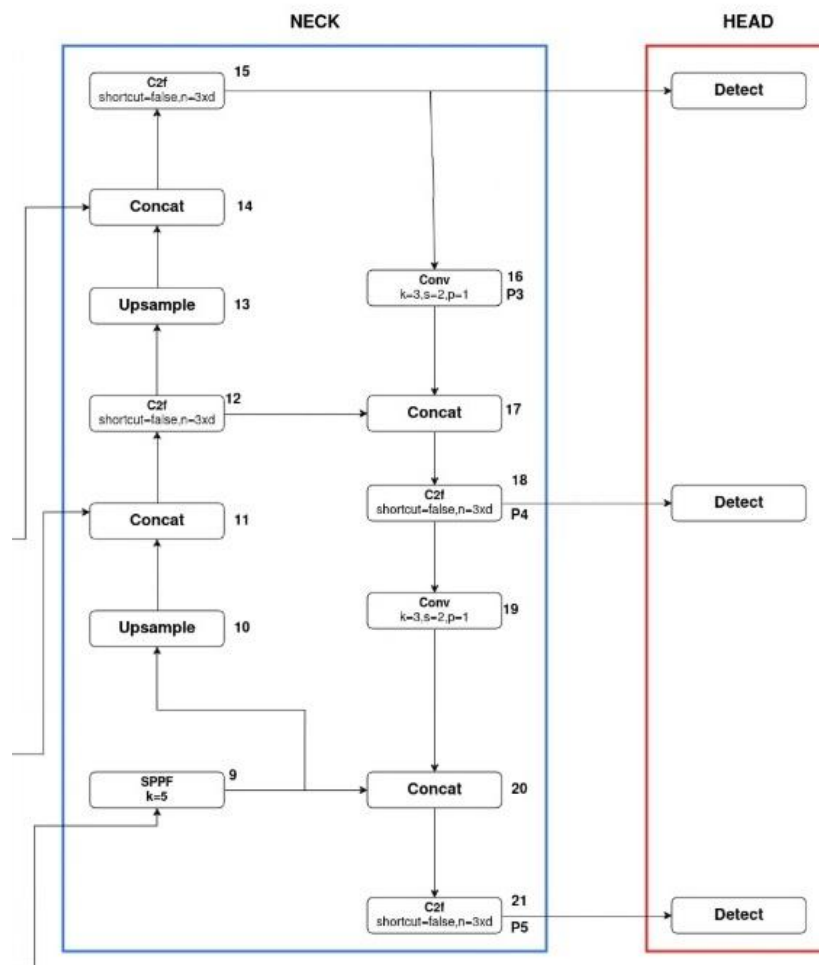


Рисунок 2.7 – Блок YOLOv8, Neck і Head.

Кожен з трьох потоків ознак, що надходять з писк, має власний шар згортки, який адаптовано до конкретного масштабу вхідних тензорів.

Кожен з трьох потоків ознак, що надходять з писк, має власний шар згортки, який адаптовано до конкретного масштабу вхідних тензорів. У відповідності до структурної логіки моделі, де кожен відповідає за певну просторову роздільність.

З технічної точки зору, кожен Detect-блок виконує одночасно три завдання: локалізацію, класифікацію та оцінку ймовірності присутності об'єкта в рамці. Ці завдання реалізуються паралельно через окремі гілки згорткових шарів, які під час інференсу повертають масив тензорів з координатними та класовими предикатами. Саме така мультигілкова структура дозволяє моделі виявляти не лише «що це» і «де це», а й оцінювати ступінь впевненості, що саме цей об'єкт заслуговує на увагу, відкидаючи потенційно хибнопозитивні зони.

Інноваційним аспектом є anchor-free механізм детекції. На відміну від попередніх реалізацій YOLO, де використовувалися попередньо визначені фрейми, YOLOv8 повністю відмовляється від цієї концепції. Замість неї застосовується точкова регресія координат яка дозволяє моделі вивчати оптимальні положення об'єктів у зображенні без необхідності ручного налаштування множини anchor-рамок. Це не лише прискорює процес тренування, а й покращує адаптацію до нових, неструктурованих датасетів [33].

Ще одним аспектом який варто наголосити, являється чітка сегментація голови на три незалежні гілки, модель зберігає високу ефективність навіть у випадках коли велика кількість об'єктів присутня в кадрі одночасно. Це дозволяє не перевантажувати окремі канали обробки, а натомість рівномірно розподіляти навантаження, що особливо важливо при роботі в реальному часі на пристроях із обмеженою обчислювальною потужністю.

Таким чином, head YOLOv8 – це високоефективна система паралельного прогнозування, яка забезпечує одночасну обробку багатьох об'єктів різного розміру без компромісів щодо швидкодії. Це досягається за рахунок продуманої модульної структури, повної відмови від anchor-підходу, використання

спеціалізованих Conv-блоків і мультигілкової організації кінцевих прогнозів. У результаті, модель не лише ефективно масштабується, але й демонструє відмінні результати у високонавантажених сценаріях, таких як розумне відеоспостереження, автономна навігація або аналіз повітряних знімків.

Якщо розглядати вище описане у більш короткій формі, то загалом архітектура YOLOv8 поєднує в собі гнучку модульну побудову з оптимізованими згортковими шарами, що дозволяє ефективно вирізняти об'єкти на зображеннях у реальному часі. Модель формує багаторівневе представлення простору ознак, використовуючи варіативні ядра згорток для балансування між точністю й швидкістю. Важливими елементами є операції підвибіркового зниження, які суттєво скорочують обчислювальну складність без втрати семантичного контексту. Заключні повнозв'язні шари дозволяють моделі прогнозувати координати рамок, клас об'єкта та ймовірність розпізнавання з високою точністю. Усе це робить YOLOv8 надзвичайно ефективним інструментом для задач комп'ютерного зору в системах з обмеженими обчислювальними ресурсами, таких як вбудовані рішення на базі БПЛА.

## **2.4 Підготовка робочих вхідних даних для системи**

Ефективне функціонування системи комп'ютерного зору, зокрема на основі архітектури YOLOv8, неможливе без чіткої та стандартизованої структури проекту.

Усі вхідні дані набору даних організовано в єдину директорію data, що містить підпапки train і valid, які в свою чергу включають по дві директорії: images та labels. Така структура дозволяє виконувати автоматизоване зчитування, генерацію батчів та логічну перевірку відповідностей без втручання користувача на кожному кроці. Усі мітки створені у форматі YOLO TXT, де кожен рядок описує один об'єкт із зазначенням класу та нормалізованих координат обмежувальної рамки.

Візуальна схема структури подана на рисунку нижче, що відображає стандартну реалізацію організації набору даних у системах типу YOLO (рисунок 2.8), [35].

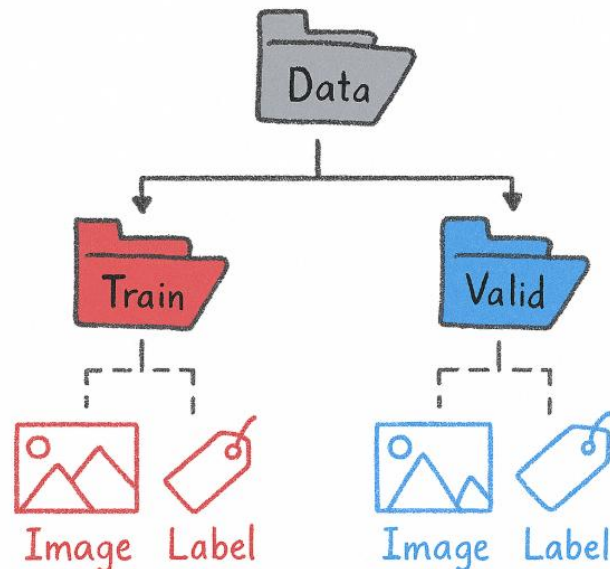


Рисунок 2.8 – Загальна структура даних, що відповідає стандарту прийнятому у YOLOv8.

Така архітектура дозволяє легко інтегрувати кілька датасетів у спільну структуру для мультикласового навчання або тонкого донавчання, забезпечуючи масштабованість і гнучкість процесу. Всі зображення та анотації мають бути синхронізовані за іменем файлу, а для управління класами та шляхами використовується окремий YAML-конфігураційний файл який описує ієрархію даних. Узагальнено, така структура дозволяє зменшити кількість помилок при підготовці навчального середовища, спрощує автоматизацію процесів і гарантує безперервність потоку даних від навчання до продуктивного застосування в реальному часі. Аналогічний підхід описано у працях присвячених конвертації з COCO у YOLO-формат, розробці hardware-aware моделей та повному циклу кастомного тренування на YOLOv8 [36], [37].

Одним з найважливіших етапів проєктування інтелектуальних систем комп'ютерного зору є вибір і формування відповідного набору даних. Для задач розпізнавання в реальному часі з повітряної перспективи необхідна наявність репрезентативних прикладів для кожної категорії об'єктів. Через суттєву

варіативність типів об'єктів, кутів огляду, масштабів, умов освітлення та фону, рішенням проєкту стало використання кількох незалежних моделей YOLOv8, кожна з яких навчена на вузькоспеціалізованому датасеті, оптимізованому під конкретну категорію об'єктів. Такий підхід дозволяє досягти кращої точності та знизити ризик хибнопозитивних спрацьовувань у складних сценах.

Загалом у межах кваліфікаційної роботи бакалавра було використано що найменше вісім окремих датасетів, з яких було обрані потрібні данні для об'єднання та підсилення варіативності в межах одного класу, тоді як інші датасети застосовувались автономно для побудови окремих моделей.

Ось мала частина датасетів для детекції транспортних засобів що були використані для навчання:

Top-View Drone Car Detection Dataset [38] – понад 12 000 зображень автомобілів з повітряної перспективи, що ідеально підходить для навчання моделей для аерозйомки. Датасет містить якісну розмітку та дозволяє моделі навчатися визначати автомобілі навіть під кутом або частково закриті перешкодами.

Саме важливе і цінне це датасети для розпізнавання людей:

YOLO HighVis and Person Detection Dataset [39] – спеціалізований набір, що містить зображення людей у світловідбивному одязі. Є критично важливим для пошукових операцій, де метою є виявлення робітників, рятувальників або цивільних осіб, одягнених у захисне спорядження;

Human Detector [40] – датасет із загальними прикладами людей у різних позах, умовах освітлення та одязі. Служив додатковим джерелом для збалансування класу «людина»;

JAAD Frames Dataset [41] – зібраний для аналізу поведінки пішоходів, цей датасет застосовано для навчання виявлення окремих постатей у складному міському середовищі.

Також не обійшлося без болючого, датасету для розпізнавання військових об'єктів

Military Assets Dataset [42] – датасет містить 12 категорій військової техніки та інфраструктури, включаючи танки, БТРи, гармати, вантажівки, будівлі та фортифікації. Формат YOLOv8 дозволив одразу використовувати його для створення спеціалізованої моделі виявлення озброєння на полі бою.

Дані було організовано згідно зі схемою, прийнятою у YOLOv8 (рисунки 2.8). Для кожної моделі було розроблено окремі директорії train/ та valid/, які містили підкаталоги images/ та labels/. Розмітка зберігалася у форматі TXT з нормалізованими координатами обмежувальних рамок.

Анотація зображень здійснювалася частково вручну. Валідація формувалася стратифіковано для забезпечення репрезентативності. Окремі моделі тренувалися автономно для підвищення спеціалізації, що дозволяє увімкнути лише потрібну модель у залежності від конкретного завдання, як от під час обльоту території у пошуках автотранспорту або під час виявлення ознак присутності людей.

## **2.5 Особливості використання спеціалізованих програмних компонентів**

Реалізація інтелектуальної системи для розпізнавання об'єктів у реальному часі з використанням безпілотних літальних апаратів вимагала залучення широкого спектра програмних бібліотек, які забезпечують необхідну функціональність на етапах підготовки, навчання, візуалізації та взаємодії з операційною системою. Особливу увагу було приділено бібліотекам, пов'язаним з глибоким навчанням, обробкою зображень та відео, побудовою графічного інтерфейсу користувача.

Для розробки методу використано низку спеціалізованих бібліотек, кожна в свою чергу виконує окрему функцію. PyTorch застосовується для запуску моделі та обробки зображень, Ultralytics є офіційною оболонкою YOLOv8, що забезпечує зручний інтерфейс до нейронної мережі. OpenCV використовується для захоплення відео попередньої обробки кадрів і відображення результатів у режимі

миттєвої обробки. NumPy забезпечує обчислення з координатами, масштабування та інші математичні операції. Для створення інтерфейсу користувача застосовано PyQt5, який дозволяє накладати вікна з інформацією поверх відео. PyDirectInput і ruwin32 реалізують взаємодію з вікнами ОС, надаючи змогу керувати зовнішніми програмами, тоді як Pillow використовується для обробки та збереження зображень.

Усі використані бібліотеки взаємодіють як єдина екосистема, тим самим забезпечуючи злагоджену роботу системи. Вони оптимізують обчислення, поєднують модель із програмною логікою, візуалізують об'єкти, визначають ймовірності та підтримують адаптивність до змін. Завдяки сумісності, наявності документації та активній спільноті, процес розробки спрощується, а система стає стабільною, масштабованою та придатною до подальшого вдосконалення.

## **2.6 Висновки до розділу 2**

У межах розділу представлено повноцінну концепцію методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт. Сформовано функціональну структуру, яка забезпечує взаємодію апаратної частини з програмними модулями, орієнтовану на обробку відеоданих із мінімальною затримкою та високою інформативністю. Архітектуру нейронної мережі реалізовано на базі YOLOv8, з урахуванням її модульності та ефективної роботи в умовах обмежених ресурсів. А також передбачено використання спеціалізованих попередньо натренованих моделей, що забезпечують автономність і адаптивність системи до різних сценаріїв застосування. Підготовку вхідних даних та побудову програмної логіки здійснено з урахуванням послідовності обробки та цілісності компонентів, які функціонують як єдина узгоджена система. Досягнуто високої швидкодії, стійкості та надійності методу в умовах практичного використання.

## **Розділ 3 Експериментальне дослідження методу та його програмна реалізація**

### **3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення**

У межах кваліфікаційної роботи бакалавра було поставлено завдання дослідити ефективність розробленого методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт з використанням глибокої згорткової нейронної мережі YOLOv8.

Для реалізації програмної частини використано стек технологій, що включає в себе як основу для обробки нейронної мережі, бібліотеку Ultralytics як інтерфейс до моделі YOLOv8, PyQt5 для генерування користувацького інтерфейсу та OpenCV для роботи з відеопотоком. Архітектура програмного забезпечення є модульною, що дозволяє тестувати окремі компоненти системи незалежно від інших, як модуль захоплення кадру, детекції об'єктів, візуалізації.

В рамках дослідження було поставлено наступні цілі:

1. Функціональне тестування – перевірка здатності системи точно виявляти об'єкти у реальному часі.
2. Тестування продуктивності – оцінка швидкодії системи: кадри за секунду, час реакції на зміни у відео, затримка між обробкою кадру та виведенням результату.
3. Тестування стабільності роботи під навантаженням – аналіз функціонування системи протягом тривалого часу без зниження якості детекції чи аварійних збоїв.
4. Тестування адаптивності до умов зйомки – визначення ефективності роботи при різному освітленні, погодних умовах, кутах огляду тощо.
5. Тестування точності – визначення ефективності визначення класу на різних відстанях відеопотоку.

Дослідження буде проводитись на ряді відео, які симулюють відеопотік із БПЛА, зокрема у форматах максимально наближених до реального польового

використання. Ці відео відображають широкий спектр умов, включаючи денне та нічне освітлення, змінну погоду, динамічний рух об'єктів, а також складні сцени з великою кількістю елементів. Завдяки цьому вдасться відтворити ситуації, що є характерними для реального бойового або пошуково-рятувального середовища, що дозволяє перевірити стійкість, точність і адаптивність системи виявлення. Особливу увагу буде приділено стабільності детекції об'єктів при нестабільному русі камери та наявності візуальних перешкод, так як це часто спостерігається під час роботи БПЛА в умовах потреби миттєвої обробки.

### **3.2 Вибір засобів розробки методу**

Розробка методу розпізнавання об'єктів у реальному часі для БПЛА у завданнях розслідувань і пошукових робіт потребує створення технічного рішення, яке має ефективно функціонувати в умовах стрімкого потоку даних та постійно змінюваних зовнішніх факторів. Основна складність полягає у потребі обробляти величезну кількість візуальної інформації із мінімальною затримкою, при цьому забезпечуючи високу точність розпізнавання об'єкту. Система повинна адекватно реагувати на різноманітні сценарії, що включають зміни освітлення, наявність перешкод, рух камери та інші фактори, які можуть впливати на якість вхідного сигналу.

Ключовим аспектом при проектуванні є досягнення оптимального балансу між продуктивністю та точністю роботи алгоритмів методу. З одного боку метод має забезпечувати швидкість обробки досить достатню для роботи у реальному часі, але з іншої сторони метод повинен підтримувати високий рівень достовірності результатів. Це досягається за рахунок ретельного підходу до архітектури рішення, де кожен компонент оптимізований для виконання своїх функцій з максимальною ефективністю. Особлива увага приділяється механізмам адаптації до змінних умов роботи, що дозволяє системі зберігати стабільність продуктивності незалежно від зовнішніх факторів.

Інтеграція системи в існуючі технологічні процеси вимагає ретельного продумування інтерфейсів взаємодії та механізмів обміну даними. Важливо забезпечити можливість безперебійної роботи в складних інфраструктурних умовах, коли необхідно обробляти дані з кількох джерел одночасно та надавати результати аналізу різним підсистемам. Гнучкість архітектури дозволяє масштабувати рішення від компактних вбудованих систем до потужних серверних конфігурацій, при цьому зберігаючи стабільність і передбачуваність роботи в будь-якому з режимів експлуатації.

Ергономіка інтерфейсу користувача відіграє вирішальну роль у забезпеченні ефективної роботи методу. Візуалізація результатів має бути інтуїтивно зрозумілою, дозволяючи оператору швидко сприймати ключову інформацію та приймати обґрунтовані рішення. При цьому метод повинен надавати достатній рівень деталізації для глибшого аналізу ситуації у випадку необхідності. Такий підхід забезпечує зручність роботи як у штатних режимах, так і в умовах підвищеної оперативної напруги.

Розробка методу розпізнавання об'єктів у реальному часі ґрунтується на комплексному підході, який поєднує в собі високопродуктивні алгоритми обробки зображень з адаптивними механізмами роботи в умовах нестабільного середовища, де ключовими пріоритетами є мінімізація затримок при збереженні точності детекції і створення інтуїтивного інтерфейсу для ефективної взаємодії оператора з системою, реалізація гнучкої модульної архітектури для подальшого масштабування та можливість безперервного вдосконалення алгоритмів без необхідності повної заміни платформи, що в сукупності забезпечує стабільну роботу в реальних умовах експлуатації при будь-якому навантаженні.

В основі системи лежить мова Python, що виступає універсальним каркасом для інтеграції всіх компонентів проєкту. Вибір саме Python обумовлений не лише її популярністю у сфері машинного навчання, а й практичними характеристиками, які відповідають вимогам до систем сучасного часу. Python дозволяє зосередитись на побудові логіки методу без глибокого занурення у керування пам'яттю або низькорівневі оптимізації, що притаманні іншим мовам.

Найважливішим компонентом методу став механізм детекції об'єктів, реалізований через модель YOLOv8 із фреймворку Ultralytics. Версія YOLOv8n була обрана як найбільш збалансована конфігурація, яка дозволяє працювати зі швидкістю понад 40 кадрів на секунду без апаратного прискорювача, що в свою чергу є критичним у випадках польового застосування без доступу до потужних серверів. Модель yolov8n оптимізована для задач із пріоритетом швидкості реагування, як от виявлення людей, тварин, різних видів транспорту в реальному часі з борта БПЛА. Хоча ця версія поступається більшим моделям у точності, але вона демонструє достатню ефективність у виявленні об'єктів середньої складності. До того ж її легко донавчати за допомогою специфічних датасетів без перебудови всієї моделі, що дозволяє підлаштовувати систему під нові сценарії пошуку.

Ще однією критично важливою складовою проєкту є каркас графічного інтерфейсу користувача, який має відображати результати розпізнавання у реальному часі поверх потоку відео. Для виконання цього завдання обрано бібліотеку PyQt5, що забезпечує підтримку прозорих оверлеїв та асинхронної обробки подій, що є вкрай важливо у контексті безперервного надходження зображень із камери БПЛА. PyQt5 дозволила не лише реалізувати швидкодійний графічний каркас інтерфейсу користувача. Завдяки візуальній простоті та можливості адаптації вікон під різні роздільні здатності, інтерфейс став зручним інструментом для візуального підтвердження виявлених об'єктів. PyQt5 був використаний на етапі прототипування, що значно прискорило розробку.

Усі вхідні дані набору даних було організовано в єдину директорію data, що містить підпапки train і valid, які в свою чергу включають по дві директорії: images та labels. Їх застосування дозволило сформувати вузько спрямовані датасети, що відображають реальні сценарії використання дронів: зйомка на відкритій місцевості, за різних погодних умов, з великої висоти. Для забезпечення швидкої обробки розмічених даних відбувся ескорт YOLO TXT, який підтримується всіма версіями Ultralytics. Його структура, яка містить лише координати об'єктів та їхні класи, дозволяє уникати перевантаження системи

метаданими. Такий підхід забезпечує максимальну продуктивність при генерації batch-box для навчання, а також мінімізує ризик помилок при інтерпретації розмітки.

Ретельно зважений вибір засобів розробки дозволив створити адаптивний та надійний метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, який має змогу автономно працювати в режимі миттєвої обробки, з каркасом інтерфейсу оператора, можливістю адаптації під нові завдання та ефективною обробкою даних у польових умовах. Кожен компонент був обраний на основі глибокого аналізу його відповідності до цільових умов застосування, а не лише на основі загальної продуктивності чи популярності.

### **3.3 Структура та функціональне призначення програмних складових системи**

У процесі розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, особливу увагу було приділено архітектурі програмного забезпечення (рисунки 3.1). Щоб візуалізувати структуру ключових компонентів, нижче побудовано діаграму класів, на якій схематично відображено взаємозв'язки між основними елементами реалізованого методу. Ця діаграма дозволяє не лише краще зрозуміти логіку проєкту, але і спростити, покращити, модифікувати, надати супровід та масштабування системи.

На поточному етапі до діаграми увійшли два класи, що формують основу користувацького інтерфейсу та реалізують логіку взаємодії між детектором і відображенням результатів. Клас `ObjectDetectionApp` виступає як керуючий модуль – він відповідає за ініціалізацію GUI-компонентів, запуск основного циклу обробки кадрів, а також за виклик методів детекції об'єктів. У цьому класі зосереджено головну логіку, яка в свою чергу забезпечує неперервний потік обробки зображення в режимі миттєвої обробки.

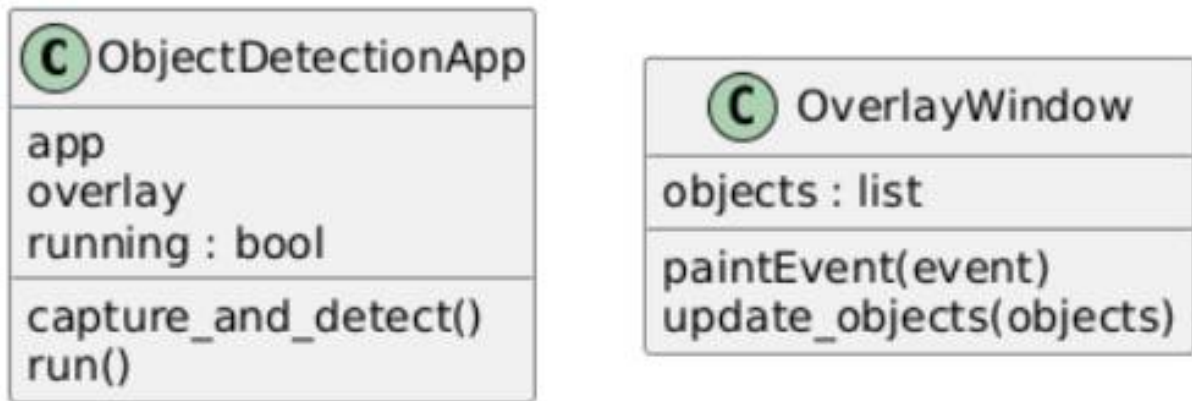


Рисунок 3.1 – Діаграма класів.

Другим ключовим елементом є клас `OverlayWindow`, який використовується у якості вікна-оверлею, накладене поверх інших застосунків. Саме тут відбувається візуалізація результатів роботи нейромережі і захоплення об'єкту, підписи класів, умовне кольорове кодування. Зміна вмісту вікна керується шляхом оновлення списку об'єктів та автоматичного перемальовування сцени при кожному новому кадрі.

Хоча діаграма виглядає маленькою та простою, но вона точно відображає принцип "мінімальної, але достатньої" архітектури, яка є характерною для прототипів у реальному часі. При цьому сам метод має модульну структуру, тому наявні компоненти можуть бути легко доповнені новими, зокрема блоками логування, керування відеопотоком, управління датасетом або додатковими панелями інтерфейсу. Представлена діаграма класів фіксує ядро застосунку, та зв'язок між детектором і візуалізацією. Вона створює фундамент, який легко масштабувати у разі подальшого розвитку системи не порушуючи загальної архітектури та логіки проекту.

### 3.4 Особливості реалізації програмних складових системи

У межах реалізації методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, важливою складовою стало донавчання нейронної мережі YOLOv8 на

спеціалізованому датасеті. Для оптимізації під реальні сценарії роботи з відео від безпілотних літальних апаратів, було обрано конфігурацію моделі YOLOv8n, яка поєднує високу швидкодію з помірними вимогами до обчислювальних ресурсів.

Навчання моделі YOLOv8n проводилось із метою створення високошвидкісної та достатньо точної нейронної мережі для детекції об'єктів у реальному часі. Вибір моделі yolo8n був зумовлений необхідністю забезпечення низького споживання ресурсів при збереженні прийнятної точності, що особливо актуально для завдань обробки відеопотоку з безпілотних літальних апаратів у польових умовах. Зазначена архітектура надає компроміс між продуктивністю та ефективністю, що робить її ідеальним кандидатом для вбудованих і десктопних рішень (рисунок 3.2).

	from	n	params	module	arguments
0	-1	1	2320	ultralytics.nn.modules.conv.Conv	[3, 80, 3, 2]
1	-1	1	115520	ultralytics.nn.modules.conv.Conv	[80, 160, 3, 2]
2	-1	3	436800	ultralytics.nn.modules.block.C2f	[160, 160, 3, True]
3	-1	1	461440	ultralytics.nn.modules.conv.Conv	[160, 320, 3, 2]
4	-1	6	3281920	ultralytics.nn.modules.block.C2f	[320, 320, 6, True]
5	-1	1	1844480	ultralytics.nn.modules.conv.Conv	[320, 640, 3, 2]
6	-1	6	13117440	ultralytics.nn.modules.block.C2f	[640, 640, 6, True]
7	-1	1	3687680	ultralytics.nn.modules.conv.Conv	[640, 640, 3, 2]
8	-1	3	6969600	ultralytics.nn.modules.block.C2f	[640, 640, 3, True]
9	-1	1	1025920	ultralytics.nn.modules.block.SPPF	[640, 640, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	3	7379200	ultralytics.nn.modules.block.C2f	[1280, 640, 3]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	3	1948800	ultralytics.nn.modules.block.C2f	[960, 320, 3]
16	-1	1	922240	ultralytics.nn.modules.conv.Conv	[320, 320, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	3	7174400	ultralytics.nn.modules.block.C2f	[960, 640, 3]
19	-1	1	3687680	ultralytics.nn.modules.conv.Conv	[640, 640, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	3	7379200	ultralytics.nn.modules.block.C2f	[1280, 640, 3]
22	[15, 18, 21]	1	8718931	ultralytics.nn.modules.head.Detect	[1, [320, 640, 640]]

Model summary: 209 layers, 68,153,571 parameters, 68,153,555 gradients, 258.1 GFLOPs

Рисунок 3.2 – Архітектурна структура моделі YOLOv8n під час ініціалізації тренування.

Для навчання використовувалась бібліотека описані раніше, однак процес навчання її можна побачити нижче (рисунок 3.3).

Вище подано комбіновану аналітичну панель, на якій представленій найбільш репрезентативний мультифункціональний датасет, що використовувався у процесі навчання моделі для виявлення об'єктів. Гістограма, розміщена у верхній лівій частині зображення, ілюструє виражену нерівномірність у розподілі класів. Спостерігається значна концентрація

прикладів для класів "person", "car", "drone", тоді як такі категорії, як "military\_vehicle", "artillery" або "aircraft", мають значно менше представництво. Подібний дисбаланс у наборі даних є типовою проблемою, що безпосередньо впливає на точність класифікації та може призводити до зміщення моделі у бік більш представлених класів, а це у свою чергу знижує загальну здатність до узагальнення.

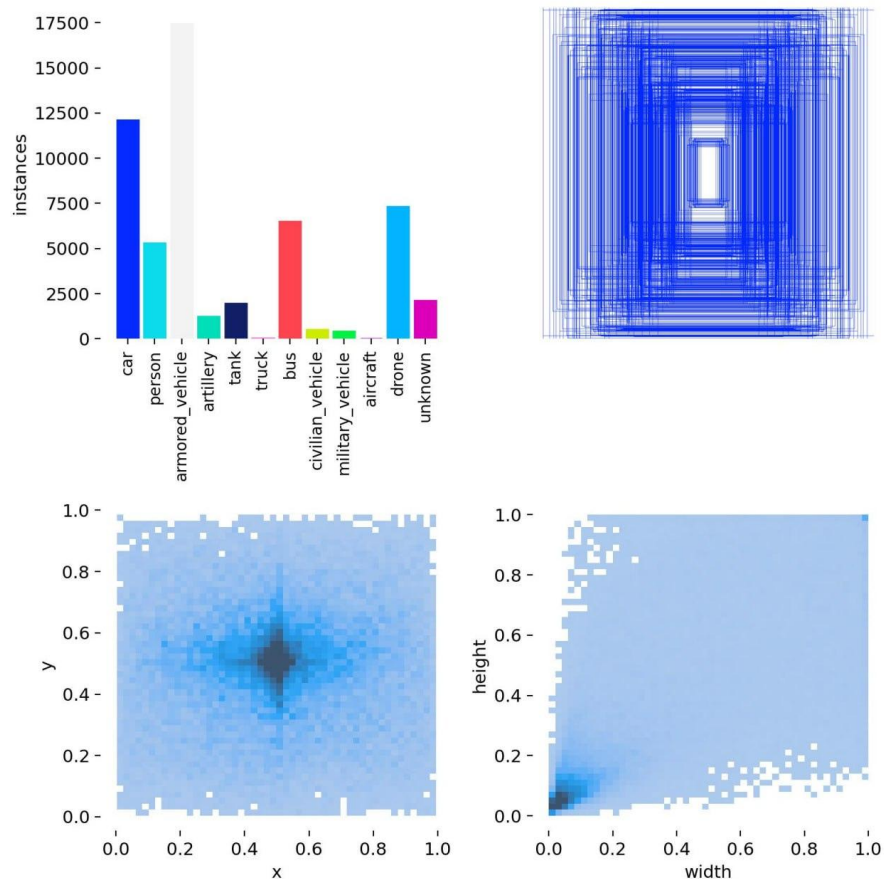


Рисунок 3.3 – Комбінована аналітична панель.

Діаграма розташована правіше демонструє накладення усіх bounding boxes у межах кадру. Такий тип візуалізації дозволяє зробити висновки щодо просторової концентрації об'єктів. У цьому випадку, є явним значне накладання прямокутників у центральній частині зображення, що свідчить про тенденцію до фіксації об'єктів поблизу центру. Це може бути пов'язано з особливостями побудови датасету, зокрема фокусуванням камер БПЛА або специфіка сцен, де найбільш релевантні об'єкти зазвичай потрапляють у центр кадру.

У нижній частині зображення розташовано ще одну ключову діаграму. Перша з них представлена як двовимірний щільнісний карт розподілу координат центрів. Вона підтверджує попередній висновок щодо централізованого розміщення об'єктів, тим самим демонструючи яскраво виражене згущення у середній області простору кадру. Це може вказувати як на специфіку сцен зйомки, так і на можливу надмірну орієнтацію розмітників на центральні об'єкти. Діаграма правіше представлена так само, але має співвідношення ширини та висоти об'єктів. Візуалізація показує високу щільність малих розмірів об'єктів, що мало займають місця кадру, це також потрібно враховувати при підборі якорів та стратегій масштабування під час навчання.

Дана статистична візуалізація не лише дозволяє глибше зрозуміти внутрішню структуру даних, а й виступає важливим діагностичним інструментом на етапі підготовки та нормалізації датасету. Правильна інтерпретація цих графіків дозволяє скоригувати модель, поліпшити баланс класів через аугментації чи переобрання даних, а також підвищити точність і надійність виявлення об'єктів у різноманітних умовах середовища (рисунок 3.3).

Рисунок нижче демонструє інтеграцію готового механізму тренування, перевірки, логування та візуалізації. Навчальний процес охоплював 100 епох при розмірі зображення  $640 \times 640$  пікселів. Паралельна обробка забезпечувалась чотирма робочими потоками, а всі обчислення виконувались на GPU з використанням CUDA.

У динаміці навчання видно (рисунок 3.4), як поступово знижується `box_loss` з 1.276 до 1.170 та `cls_loss` з 1.350 до 1.128 між 7 та 12 епохами, а значення `mAP@50` зростає з 0.424 до 0.528. Це демонструє прогрес у здатності моделі правильно локалізувати об'єкти на зображеннях та їх класифікувати. Крім того, використання `dfl_loss` дозволяє моделі краще регресувати координати обмежувальних рамок, що критично важливо для миттєвої обробки.

Epoch	GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	640:	100%	2944/2944	[29:54<00:00,
7/100	6.65G	1.276	1.35	1.479	19	640:	100%	2944/2944	[29:54<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.565	0.416	0.424	0.265			
8/100	6.65G	1.24	1.284	1.458	13	640:	100%	2944/2944	[30:04<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.549	0.446	0.45	0.284			
9/100	6.64G	1.22	1.234	1.439	31	640:	100%	2944/2944	[29:42<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.69	0.404	0.468	0.296			
10/100	6.67G	1.206	1.194	1.428	16	640:	100%	2944/2944	[29:44<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.657	0.437	0.473	0.305			
11/100	6.64G	1.183	1.163	1.416	15	640:	100%	2944/2944	[29:42<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.617	0.456	0.489	0.312			
12/100	6.62G	1.17	1.128	1.405	31	640:	100%	2944/2944	[29:43<00:00,	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	212/212	[04:
	all	3389	8523	0.638	0.487	0.528	0.344			

Рисунок 3.4 – Динаміка навчання моделі YOLOv8n за ключовими метриками втрат і точності.

Навчання завершувалось створенням двох файлів `best.pt` та `last.p`, це забезпечує використання самої кращої моделі, або продовжити покращувати, все результати включно із вище описаними зберігаються у відповідному каталозі експериментів, що дозволяє повторно аналізувати ефективність тренування, порівнювати з альтернативними конфігураціями та створювати інтерактивні візуалізації.

Таким чином реалізований процес навчання створив компактний, оптимізований набір даних, який успішно поєднує продуктивність та адаптивність і високу швидкодію на побудові GPU та готовий до інтеграції в повноцінну систему із детекцією об'єктів з БПЛА в умовах обмеженого доступу до обчислювальних ресурсів.

Однією з ключових особливостей реалізованої методу є підтримка візуального відображення результатів розпізнавання об'єктів із миттєвою обробкою, яка реалізована з використанням бібліотеки `PyQt5`. Завдяки цьому було досягнуто максимальної інтеграції з робочим середовищем користувача без необхідності перемикавання між додатками або використання спеціального вікна перегляду. Графічний модуль системи представлено у вигляді оверлейного прозорого вікна, що накладається поверх усіх інших вікон операційної системи.

Вікно не містить жодного стандартного оформлення та має повністю прозоре тло, що забезпечується за допомогою встановлення спеціальних прапорців. Завдяки цьому підхід є незалежним від типу запущених застосунків та дозволяє інтегрувати метод візуалізації з вхідним відеопотоком, отриманим із БПЛА або з інших джерел.

Під час обробки кадру результати розпізнавання передаються до модуля візуалізації. Для кожного виявленого об'єкта формується прямокутник з чітким кольоровим контуром, а також напівпрозоре тло з підписом класу. Відображення адаптоване під різні роздільності екрана, автоматично масштабуючи елементи при зміні розміру дисплея.

Під час реалізації було передбачено обробку множини об'єктів з високою частотою оновлення, що дозволяє підтримувати стабільну візуалізацію навіть при низькому FPS. Особливу увагу приділено відображенню текстової інформації, де використовується жирний шрифт на темному фоні для підвищення контрастності, а також алгоритмічне уникнення накладання тексту на графіку інших об'єктів. Нижче наведено приклад роботи системи де відображено виявлені об'єкти, які накладаються у реальному часі поверх будь-якого активного вікна на екрані (рисунок 3.5).

Цей підхід має перевагу над класичними OpenCV-вікнами, оскільки дозволяє вільно налаштовувати візуальний інтерфейс і використовувати прозорість, і тим самим забезпечувати інтеграцію в операційне середовище без втрати контексту користувача. Завдяки цьому система є придатною для використання в польових умовах, включаючи сценарії з обмеженою увагою оператора або під час керування БПЛА в реальному часі.

У результаті реалізації програмного забезпечення було розроблено функціонально завершений метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, який орієнтований на виявлення та класифікацію об'єктів у відеопотоці в реальному часі. Архітектура застосунку побудована з урахуванням вимог до високої продуктивності, масштабованості та мінімального споживання ресурсів.



Рисунок 3.5 – Виявлення та візуалізація об’єкта в методом розпізнавання об’єктів у реальному часі.

Основою системи є модель YOLOv8, яка інтегрується в прикладне середовище через бібліотеку Ultralytics. Модульна побудова дозволяє чітко відокремити етапи захоплення зображень, обробки, інтерпретації результатів та візуалізації. Особливістю реалізації є поєднання сучасних бібліотек для обробки відео, управління каркасом графічного інтерфейсу, низькорівневої взаємодії з ОС та математичних обчислень. Такий підхід дозволив реалізувати прозоре оверлейне вікно поверх сторонніх програм, а також забезпечити коректну взаємодію з іншими процесами системи без потреби у додатковому програмному забезпеченні.

Підтримка форматів експорту забезпечує можливість подальшого перенесення моделі на мобільні або вбудовані пристрої. Загалом реалізований метод відповідає критеріям надійності, зручності та гнучкості у використанні для майбутнього розширення функціоналу, інтеграцію з зовнішніми джерелами даних або впровадження розширених систем логуювання та моніторингу.

### 3.5 Аналіз функціональності системи

Для запуску системи автоматизованого виявлення об'єктів користувачу необхідно відкрити термінал, або командний рядок перейшовши до каталогу де збережено файл програми та виконати команду запуску за допомогою Python, `python main.py`

Альтернативно, у випадку налаштованого середовища метод можна запустити подвійним натисканням на файл `main.py`, якщо у системі асоційовано відкриття `.py` файлів через Python-інтерпретатор.

Після запуску користувач автоматично бачить вікно з відеопотоком з камери або з тестового відеофайлу, якщо відповідні параметри встановлено у конфігураціях. Детекція об'єктів відбувається в режимі миттєвої обробки, де об'єкти обводяться прямокутниками з підписами класів та відображенням точності розпізнавання. Каркас інтерфейсу не вимагає взаємодії через кнопки: система повністю автоматизована та одразу розпочинає аналіз відео після активації.

У разі потреби зупинки роботи, є можливість користувача закрити вікно або натиснути клавішу Q, яка реалізована як сигнал для завершення процесу. Усі результати логуються записано у спеціальну директорію `logs/` у форматі `.log`, що дозволяє здійснювати подальший аналіз роботи системи.

Нижче зображено приклад роботи додатку після запуску. Вікно з накладеним оверлеєм і підписаними об'єктами на фоні відеопотоку (рисунок 3.6).



Рисунок 3.6 – Вікно системи із детектованим об'єктом.

Таким чином, користування програмою не потребує попередніх технічних знань, а всього достатньо мати встановлене середовище Python, необхідні залежності та пристрій захоплення відео. Інтуїтивність, автоматичність та мінімальні вимоги до взаємодії з користувачем роблять систему зручною та ефективною в експлуатації, зокрема для задач які потребують миттєвої обробки на мобільних або польових станціях.

### 3.6 Результати досліджень

Для максимальної перевірки якості реалізованого методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт, було проведено серію тестувань, які спрямовані на аналіз функціональності, продуктивності, адаптивності та стійкості до змін зовнішніх умов. Кожне з тестувань фокусується на певному аспекті роботи системи, даючи змогу оцінити її практичну придатність до реального застосування. Дослідження виконувались із фіксованим набором параметрів, а результати документувались для подальшої інтерпретації та порівняння.

Першим тестом буде функціональне, основною метою тесту було перевірити здатність системи точно виявляти об'єкти у відеопотоці в реальному часі. Застосовувались контрольні зразки з відомими мітками об'єктів, що дозволило оцінити правильність спрацьовування моделі.

Крім контрольованих даних, у тестуванні використовувався спеціалізований датасет, що сформований на основі відеоматеріалів із реальних бойових зон. Зокрема, були обрані відео, у яких відбувається фіксувалося переміщення військової техніки в умовах зруйнованої інфраструктури населеного пункту. Такі складні умови, як пил, дим, обмежена видимість, схожі за формою та кольором об'єкти суттєво ускладнювали задачу точного виявлення. Особлива увага приділялася розпізнаванню бойової одиниці M2 Bradley, що переміщувалася поміж будівлями. Незважаючи на розмитість кадрів та візуальні перешкоди, система успішно виявила транспортний засіб. Надалі об'єкт залишався

зафіксованим без втрати трекінгу, що свідчить про надійність роботи моделі навіть у критичних ситуаціях (рисунок 3.7).

На ілюстрації нижче показано приклад виявлення транспорту біля будівель і попри складні візуальні умови, алгоритм коректно класифікував техніку, що створює передумови для її використання в системах тактичного аналізу та корегування дій у реальному часі.



Рисунок 3.7 – Функціональне тестування.

Окрім функціонального дослідження проведено ще декілька тестувань розробленого методу. Наступним тестуванням є перевірка продуктивності та стабільність роботи Ці випробування аналогічно попередньому проводилися в умовах, що є наближеними до реального середовища експлуатації з метою максимально об'єктивного вимірювання ключових параметрів ефективності.

Результат продуктивності відображено на узагальнюючій діаграмі, де представлено середні значення часу виконання окремих етапів обробки зображення, а також відповідні стандартні відхилення (рисунок 3.8). Було зафіксовано, що етап попередньої обробки виконується в середньому за 22,5 мс зі стандартним відхиленням 4,3 мс. Інференс, тобто безпосереднє розпізнавання об'єктів неймережею потребує приблизно 5,5 мс у середньому при стандартному відхиленні 1,5 мс. Відображення результатів і побудова bounding boxes займає близько 3 мс, з відхиленням 1 мс. Сукупно весь цикл обробки зображення залишається в межах, які в свою чергу дозволяють забезпечити миттєву обробку без відчутної затримки.

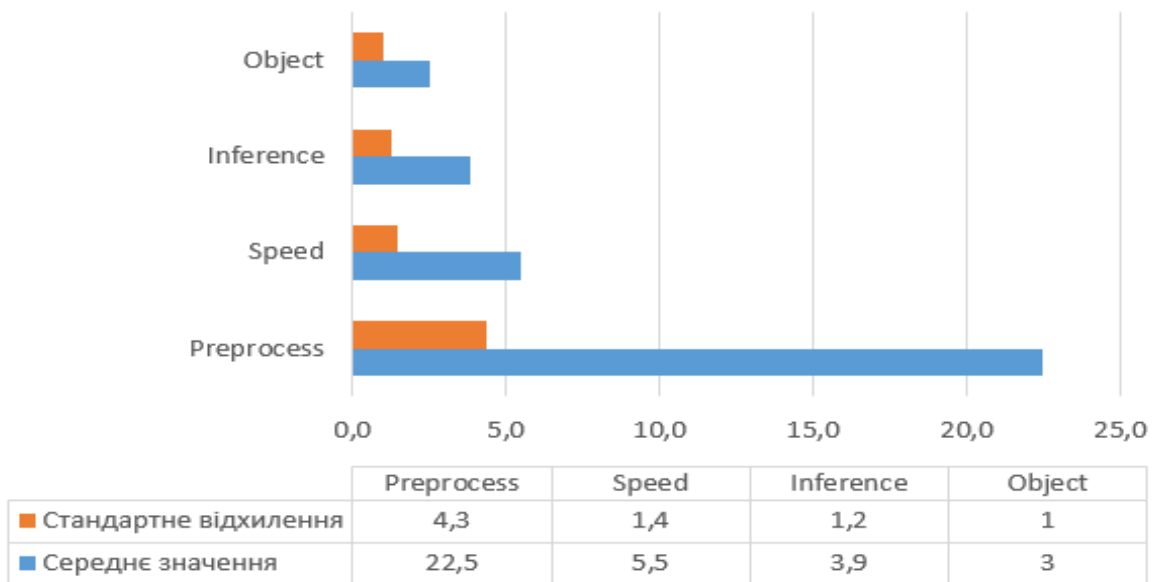


Рисунок 3.8 – Динаміка продуктивності методу.

Залежність частоти кадрів від кількості об'єктів у сцені демонструє стабільність роботи системи. У ході аналізу стало відомо, що при відсутності об'єктів швидкодія системи досягає максимального значення в 40-41 кадр/с, у той час як при наявності трьох або більше об'єктів вона знижується до рівня 18-20 кадрів/с. Зменшення FPS відбувається поступово: за умови одного об'єкта система працює в межах 28-31 кадр/с, а при двох-трьох у межах 25-28 кадр/с. Незважаючи на зростання навантаження, система демонструє стабільну адаптивність і підтримує прийнятний рівень обробки, що дозволяє зберігати якість детекції без критичних втрат продуктивності (рисунок 3.9).

Одним із складних тестів є адаптивність до умов зйомки, так як її здатність од розпізнавання впрі всіх умовах може багато що змінити. Зважаючи увага, що об'єкти детекції можуть перебувати у будь-яких обставинах, як у відкритому полі, так і в урбанізованому середовищі під впливом перешкод, відповідно є необхідність забезпечити стабільність розпізнавання незалежно від зовнішніх чинників.

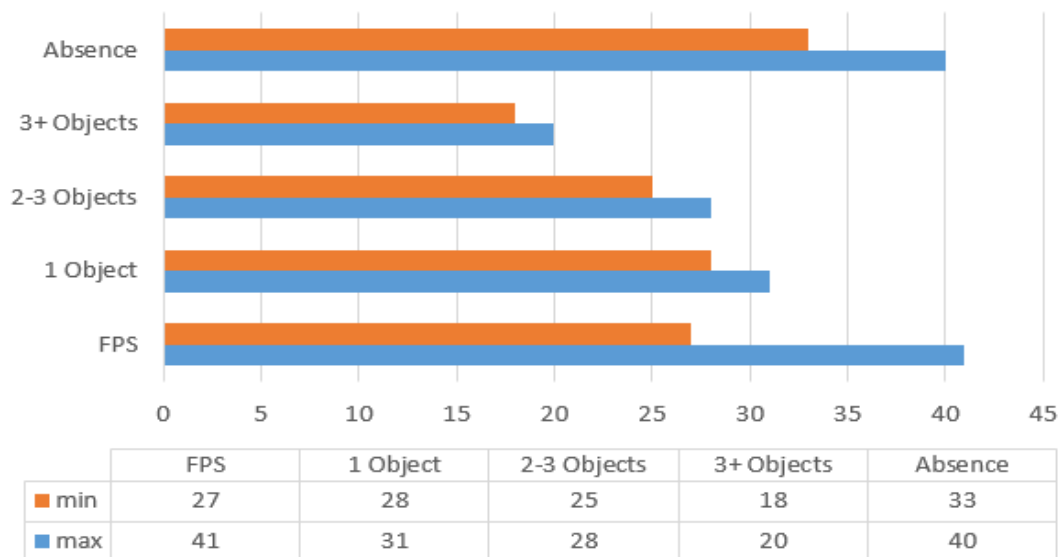


Рисунок 3.9 – Залежність FPS від кількості об’єктів у сцені.

З метою оцінки ефективності роботи розробленої системи в умовах, наближених до реальних сценаріїв, було проведено тестування із використанням відеофрагментів у рамках розслідування поліцейських справ. Такі матеріали часто мають обмежену якість, відзначаються низьким контрастом, значним рівнем шуму та нестабільною зйомкою, що створює сприятливі умови для перевірки стійкості алгоритмів комп’ютерного зору. Обраний відеоряд також ілюструє ситуацію у якій було необхідно здійснити пошук підозрюваної особи на основі доступного відеозапису. На нижче наведено приклад одного з кадрів, де модель виявила об’єкт класу person, що переміщається вздовж інфраструктурного об’єкта. Незважаючи на розмитість зображення та неповне освітлення, модель змогла ідентифікувати людину з достатньою впевненістю, що свідчить про її здатність працювати в умовах поганої видимості. Цей експеримент додатково підтверджує можливість використання запропонованої системи у сфері правоохоронної діяльності, де часто трапляються відеодокази з камер спостереження або портативних пристроїв із нестабільним потоком даних (рисунок 3.10).

Додатковим підтвердженням здатності системи розпізнавання адаптуватися до реалій природного середовища є результати обробки відеоданих, що отримані у зоні лісної або частково залісної місцевості. Нижче

представлено приклад детекції об'єктів класу vehicle, що розміщені в посадці, частково перекриті елементами рослинності та ландшафту (рисунок 3.11).

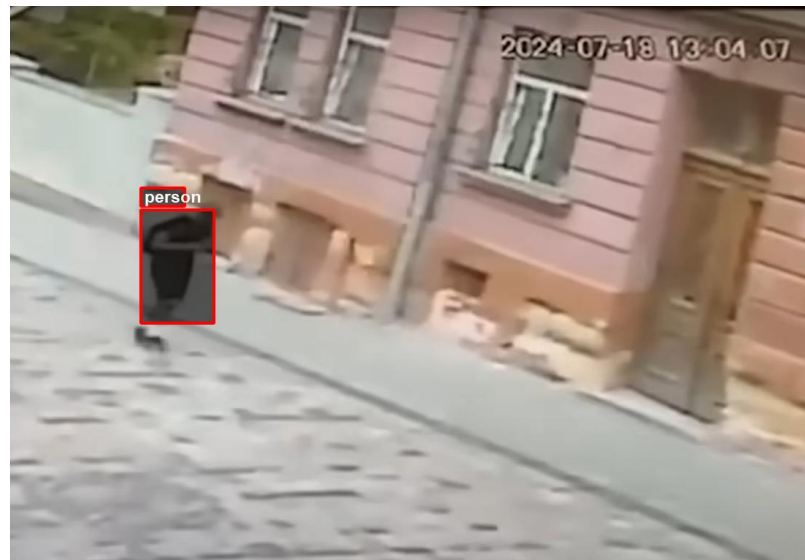


Рисунок 3.10 – Виявлення об'єкту person.



Рисунок 3.11 – Техніка у посадці.

Сцена демонструє транспортний засіб, який прихований у кронах дерев з обмеженим видимим контуром. Незважаючи на складність фону, модель успішно ідентифікує транспортну техніку завдяки глибокому аналізу просторових ознак, навчання на релевантному датасеті та використанню високої роздільності входу. Такі ситуації є типовими для пошуково-розвідувальних місій, де об'єкти можуть навмисно маскуватися у природному середовищі.

Останнім етапом тестування стала перевірка точності методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт. Основна увага зосереджувалась на визначенні рівня коректного розпізнавання класу об'єкта на різних відстанях від

5 до понад 100 метрів, а також на частоту зміни класу в умовах використання датасетів, які в свою чергу були натреновані на розпізнавання кількох класів одночасно (рисунок 3.12).

У результаті тестування було визначено середній рівень точності класифікації об'єкта залежно від відстані до нього, а також встановлено максимальне та мінімальне значення похибки, на основі аналізу 10 000 кадрів відеопотоку. Отримані результати демонструють чітку залежність між збільшенням відстані та втратою точності відбувається поступове зменшення впевненості моделі у класифікації, а також зростання кількості помилок розпізнавання та ефективності визначення класу на різних відстанях відеопотоку.

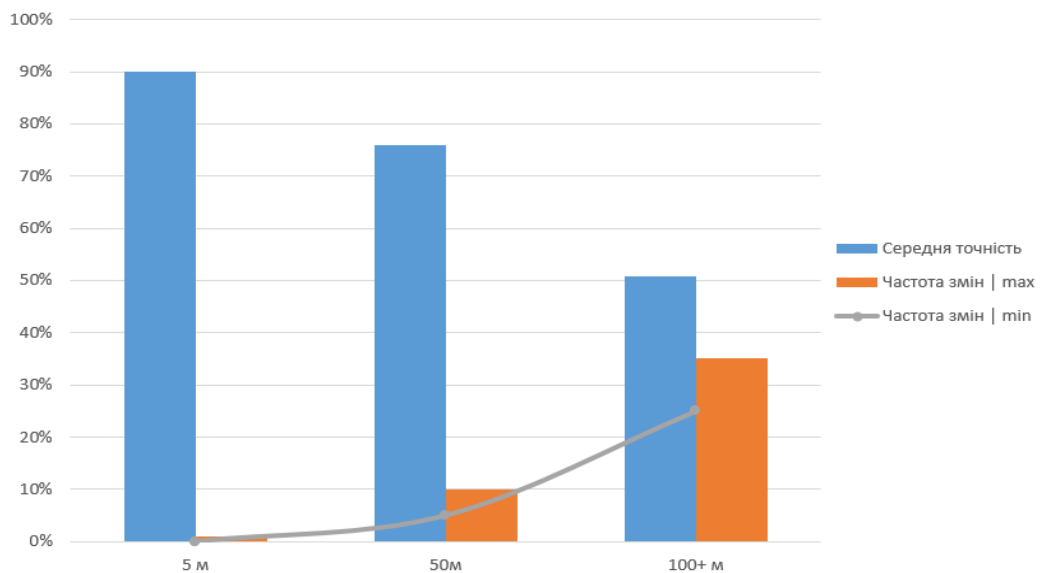


Рисунок 3.12 – Показники імовірність належності знайденого об'єкта до певного класу.

Одержані результати експериментального тестування продемонстрували високу ефективність запропонованого методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт. Проведені випробування засвідчили стабільну роботу системи в умовах змінного освітлення, наявності складного фону, великої кількості об'єктів у кадрі та динамічних сцен. Модель показала стійкість до хибних спрацювань, а також здатність до функціонування у тривалому безперервному режимі без деградації

продуктивності, можливість обробляти одразу декілька об'єктів. Ефективність моделі до розпізнавання показала досить високу точність та залежність відстані та її ефективність.

Швидкодія системи зберігається навіть при зростанні кількості об'єктів, що є критично важливим для використання у реальному часі на обмежених обчислювальних ресурсах. Надійне виявлення транспортних засобів, пішоходів та військової техніки в різноманітних умовах вказує на універсальність обраного підходу. Тому враховуючи результати тестування, потрібно підтвердити, що запропонована інформаційна система може бути з успіхом застосована у сферах, що потребують швидкої візуальної аналітики: аеророзвідка, оперативне реагування під час надзвичайних ситуацій, моніторинг дорожньої інфраструктури, забезпечення безпеки масових заходів, виявлення порушень на промислових об'єктах, тощо. Гнучкість архітектури системи також дозволяє адаптувати її до інших задач комп'ютерного зору з мінімальними зусиллями.

### **3.7 Висновки до розділу 3**

Використовуючи інструмент для проведення експериментів у вигляді розробленого методу розпізнавання об'єктів у модельованому середовищі відеопотоку для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт. Вірний підбір технологій, архітектурних рішень і програмних бібліотек дозволив досягти балансу між швидкістю, точністю та стабільністю.

Метод реалізовано на основі архітектури YOLOv8, що відзначається високою ефективністю в задачах детекції, з повним циклом навчання та валідації за допомогою бібліотеки Ultralytics і подальшою інтеграцією в середовище обробки відеопотоків у режимі, наближеному до реального часу. Розроблено каркас PyQt5-інтерфейсу із прозорим накладанням на робочу поверхню для зручної взаємодії з системою, а також впроваджено механізм живого логування у консолі, що дозволяє відстежувати події, параметри продуктивності та службові

повідомлення безпосередньо під час роботи, не вдаючись до аналізу зовнішніх журналів.

Під час тестування система показала середню точність розпізнавання на рівні 90% на відстані до 5 м, 60% – на 50 м і 50% – на 100+ м, що демонструє її ефективність для ближнього та середнього радіуса дії. Коливання результатів були незначними, так як частота змін точності не перевищувала 1% на 5 м, 10% на 50 м і 35% на 100+ м.

Тестування продуктивності дало такі результати:

- середній час препроцесингу кадру сягав 5,5 мс із відхиленням  $\pm 4,3$  мс;
- середній час інференсу сягав 3,8 мс із відхиленням  $\pm 1,4$  мс;
- час обробки одного об'єкта сягав 3 мс із відхиленням  $\pm 1$  мс.

Показники частоти кадрів також свідчать про ефективність:

- максимальне значення FPS досягало 41 кадру, а мінімальна в свою чергу досягало 27 кадрів;
- при виявленні 1 об'єкту FPS коливався у межах 31-28 кадру;
- при виявленні вже 2–3 об'єктів, FPS зменшувався 28–25 кадрів;
- при виявленні 3 чи більше об'єктів, кадри зменшувалися до: 20–18;
- при повній відсутності об'єктів було отримано стабільні 40–33 кадри.

Запропоноване рішення є масштабованим, гнучким і адаптивним, власне його модульна структура дозволяє легко інтегрувати нові джерела даних, доповнювати інтерфейс або впроваджувати додаткові алгоритми після обробки. Система успішно пройшла функціональні та продуктивні тестування, зокрема за показниками точності позиціонування, точності, та адаптації до складних умов навколишнього середовища. Отримані результати підтверджують доцільність її застосування у сферах із необхідністю швидкої й надійної візуальної аналітики як от: розслідування, пошукових робіт, військова розвідка, рятувальні операції, спостереження за інфраструктурою, забезпечення громадської безпеки тощо.

## Висновок

У результаті виконання кваліфікаційної роботи бакалавра було успішно досягнуто поставленої мети, а саме підвищення ефективності пошукових робіт і розслідувань шляхом розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів. Також успішно опрацьовано технології комп'ютерного зору, що лежать в основі розпізнавання об'єктів у межах пошукових і розслідувальних завдань із використанням безпілотних літальних апаратів, із фокусом на реалізацію та практичне застосування методу. У ході розробки методу було виконано наступний ряд завдань кваліфікаційної роботи бакалавра: проведено аналіз актуальності з завдання розробки методу; наведено дослідження щодо наявних підходів щодо предметної області цього методу; розроблено метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт; розроблено алгоритм виявлення об'єктів на відеопотоці, що дозволяє реалізувати даний метод; провести тестування програмної системи; провести експериментальні дослідження ефективності розробленого методу.

У підсумку, розроблений метод є сучасним технічним рішенням, яке поєднує в собі інтелектуальний аналіз, простоту використання, високу швидкодію та технологічну відкритість. У ході досліджень система стабільно функціонувала без перевантаження апаратних ресурсів навіть за умови обробки декількох об'єктів одночасно. Ефективність методу підтверджує його практичну доцільність для виконання поставлених завдань навіть за умов низької якості вхідного зображення чи нестабільного освітлення. Метод демонструє гнучкість до подальшої адаптації: можливе інтегрування в, криміналістичні операції, системи автоматизованих спостережень або пошуково-рятувальних операцій. У перспективі майбутнє удосконалення алгоритмів, розширення набору розпізнаваних класів, оптимізація під мобільні платформи, а також створення веб-інтерфейсу або повноцінного застосунку, що дозволить масштабувати розроблену систему в умовах реального промислового чи польового розгортання.

## Перелік посилань

1. Mukachevo.net. У Карпатах за допомогою дрона відшукали трьох туристів, які вперше вирушили у похід. 2025. URL: [https://mukachevo.net/news/u-karpatakh-za-dopomohoiu-drona-vidshukaly-tryokh-turystiv-iaki-vpershe-vyrushyly-u-pokhid\\_6273757.html](https://mukachevo.net/news/u-karpatakh-za-dopomohoiu-drona-vidshukaly-tryokh-turystiv-iaki-vpershe-vyrushyly-u-pokhid_6273757.html)
2. Framework for smartphone-based grape detection and vineyard management using uav-trained AI / S. Vélez et al. Heliyon. 2025. P. e42525.
3. A uav-based ignition detection in prescribed fires using deep learning / F. Marques et al. Icfbr 2022. Basel Switzerland, 2022.
4. ADDITIVE TECHNOLOGIES AND PROFILING OF GENERAL PURPOSE DRONES: THE FUTURE OF AVIATION / A. Vlasov et al. Reliability & Quality of Complex Systems. 2023. No. 4.
5. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends / S. A. H. Mohsan et al. Intelligent Service Robotics. 2023.
6. Carvajal-Ramírez F., Agüera-Vega F., Martínez-Carricondo P. Editorial for Special Issue “UAV Photogrammetry and Remote Sensing”. Remote Sensing. 2021. Vol. 13, no. 12. P. 2327.
7. Hussein Hasan R., Majid Hassoo R., Salman Aboud I. Yolo Versions Architecture: Review. International Journal of Advances in Scientific Research and Engineering. 2023. Vol. 09, no. 11. P. 73–92.
8. Opencil-pytorch: an OpenCL-based extension of PyTorch / Y. Sui et al. CCF Transactions on High Performance Computing. 2024.
9. Rahim A., Yuan F., Barabady J. An Ultralytics YOLOv8-Based Approach for Road Detection in Snowy Environments in the Arctic Region of Norway. Computers, Materials & Continua. 2025.
10. Ch OpenCV for interactive open architecture computer vision / Q. Yu et al. Advances in Engineering Software. 2004. Vol. 35, no. 8-9. P. 527–536.

11. Willman J. Overview of PyQt5. Modern PyQt. Berkeley, CA, 2020. P. 1–42.
12. Ветерани бойових дій. Теорія і практика застосування БПЛА. 2022. Ст 127.
13. Singh M. D. Virtual Mouse using OpenCV. International Journal for Research in Applied Science and Engineering Technology. 2021. Vol. 9, no. 12. P. 1055–1058.
14. UAV Formation Occupying Technology Research Collaborative / Y. Li et al. International Journal of Control and Automation. 2015. Vol. 8, no. 1. P. 263–274.
15. AV-Based Remote Sensing. MDPI, 2018.
16. Chary P. S. Real Time Object Detection Using YOLOv4. International Journal for Research in Applied Science and Engineering Technology. 2023. Vol. 11, no. 12. P. 1375–1379.
17. Shorten C., Khoshgoftaar T. M. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 2019. Vol. 6, no. 1.
18. Zeng W. Image data augmentation techniques based on deep learning: A survey. Mathematical Biosciences and Engineering. 2024. Vol. 21, no. 6. P. 6190–6224.
19. Applications of UAV in the agriculture field / R. Akshay et al. Applied and Computational Engineering. 2023. Vol. 6, no. 1. P. 546–551.
20. GeoDeep Project. GitHub Repository. URL: <https://github.com/uav4geo/GeoDeep>
21. GeoDeep Project. Releases. URL: <https://github.com/uav4geo/GeoDeep/releases>
22. Zhang, P., et al. SlimYOLOv3: Narrower, Faster, and Better for Real-Time UAV Applications. arXiv, 2019.
23. Zhang P., Zhong Y., Li X. SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 27–28 October 2019. 2019.

24. ScienceDirect. Advances in UAV Technology. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0168169924007336>
25. MFNet: A Novel GNN-based Multi-Level Feature Network with Superpixel Priors / S. Li et al. IEEE Transactions on Image Processing. 2022. P. 1.
26. arXiv. Advances in Object Detection. URL: <https://arxiv.org/html/2501.05885v1>
27. Improvement of Deep Learning Models to Predict the Knowledge Level of Learners based on the EdNet Data / S. Choi et al. Journal of KIISE. 2021. Vol. 48, no. 12. P. 1335–1342.
28. Wang M., Zhu S., Zhang Y. FastDet: Providing faster deterministic transmission for time-sensitive flows in WAN. Computer Networks. 2024. P. 110881.
29. Medium. An Introduction to Detectron2. URL: <https://medium.com/@ravina.lad01/an-introduction-to-detectron2-1b15e2f39b19>
30. Cricket Shot Classification and Pose Correction using Detectron2 and XGBoost Classifier / G. Majumder et al. Procedia Computer Science. 2025. Vol. 258. P. 1274–1284.
31. Comprehensive analysis on Ultralytics-supported YOLO models for detection and recognition of large office objects for indoor navigation / R. G. Baldovino et al. Procedia Computer Science. 2024. Vol. 246. P. 3851–3858.
32. Vehicle Classification and Counting System Using YOLO Object Detection Technology / J.-D. Wu et al. Traitement du Signal. 2021. Vol. 38, no. 4. P. 1087–1093.
33. FM-YOLO Object Detection Algorithm / X. Li et al. DSDE 2022: 2022 the 5th International Conference on Data Storage and Data Engineering, Sanya China. New York, NY, USA, 2022.
34. Cavin J. D. Advances in distributed sensor technology. IEEE Parallel & Distributed Technology: Systems & Applications. 1996. Vol. 4, no. 4. P. 83–84.
35. Verdhan V. Object Detection Using Deep Learning. Computer Vision Using Deep Learning. Berkeley, CA, 2021. P. 141–185.

36. Huang S.-C., Le T.-H. Object detection. Principles and Labs for Deep Learning. 2021. P. 283–331.
37. Ultralytics Documentation. COCO Dataset. URL: <https://docs.ultralytics.com/datasets/pose/coco/#sample-images-and-annotations>
38. Kaggle. Top View Drone Car Detection Dataset. URL: <https://www.kaggle.com/datasets/glebkuzntesov/top-view-drone-car-detection-dataset-12000-images>
39. Kaggle. YOLO High-Vis and Person Detection Dataset. URL: <https://www.kaggle.com/datasets/tudorhirtopanu/yolo-highvis-and-person-detection-dataset>
40. Kaggle. Human Detection Dataset. URL: <https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset>
41. Kaggle. JAAD Frames Dataset. URL: <https://www.kaggle.com/datasets/charvik/jaad-frames-dataset-10-v1-yolo-format>
42. Kaggle. Military Assets Dataset. URL: <https://www.kaggle.com/datasets/rawsi18/military-assets-dataset-12-classes-yolo8-format>

# ДОДАТКИ

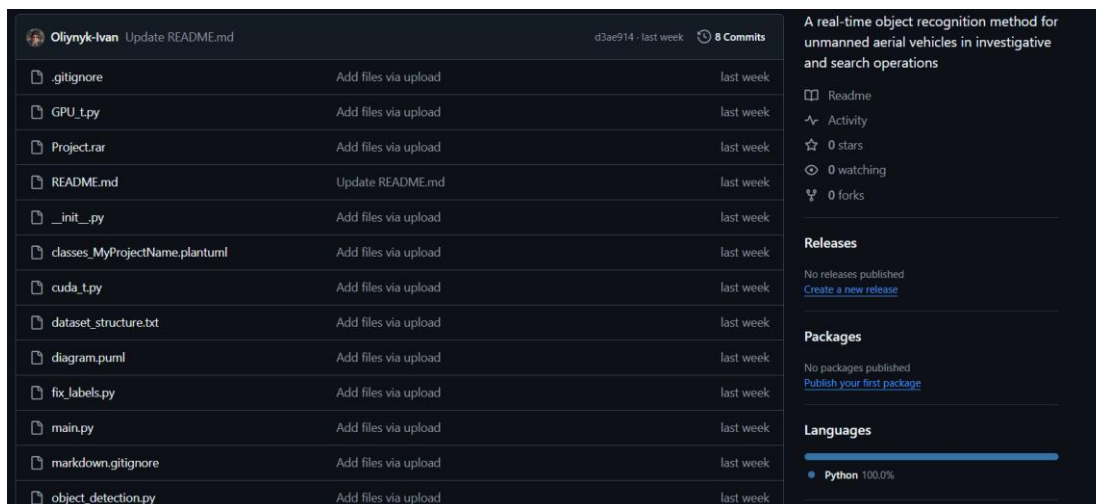
## Додаток А

### Програмний код

#### Посилання на репозиторій на GitHub:

[https://github.com/Oliynyk-Ivan/Project\\_Yolo\\_AI](https://github.com/Oliynyk-Ivan/Project_Yolo_AI)

#### Вигляд сторінки репозиторію



#### Опис вмісту

– main.py – головний файл запуску програми; відповідає за ініціалізацію та інтеграцію компонентів системи.

– object\_detection.py – модуль для роботи з моделлю YOLO, виконує детекцію об’єктів у відеопотоці.

– overlay\_window.py – відповідає за створення прозорого вікна поверх додатків.

– train.py/train\_model.py/train\_model\_1.py – скрипти для навчання нейромережі на власних датасетах, з різними конфігураціями.

– cuda\_t.py/GPU\_t.py – перевірка доступності CUDA і GPU для обчислень.

– yolov8n.pt/yolov8m.pt/yolov8x.pt/yolo11n.pt – готові моделі YOLO різних розмірів.

– yolov8n\_trained.pt – Маленька натренована моделька для перевірки навчання.

– \_\_init\_\_.py – службовий файл для оголошення пакету Python.

– .plantuml – текстові файли для побудови діаграм класів і пакетів у PlantUML.

## Додаток Б

### Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# МЕТОД РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У РЕАЛЬНОМУ ЧАСІ ДЛЯ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ У ЗАВДАННЯХ РОЗСЛІДУВАНЬ І ПОШУКОВИХ РОБІТ



**Виконав:**

*студент 4 курсу, групи КН-21-2*

**Олійник** Іван Анатолійович



**Керівник:**

*д.т.н., проф. каф. КН*

**Манзюк** Едуард Андрійович

2

## Актуальність

Сучасне зростання ролі БПЛА у цивільній, оборонній та рятувальній сферах зумовлює нагальну потребу в ефективному виявленні та розпізнаванні об'єктів у потоці відеоданих із мінімальною участю оператора. Безпілотники дедалі частіше застосовуються для моніторингу, пошуку, оцінки ситуацій у реальному часі, що вимагає швидкої та точної обробки інформації. В умовах обмежених ресурсів на борту перспективним є підхід, за якого відеопотік передається на наземну обчислювальну станцію, де за допомогою нейронних мереж родини YOLO, зокрема YOLOv8, виконується глибинний аналіз зображень. Це забезпечує високу точність виявлення без необхідності модифікації апаратного забезпечення БПЛА, що є ключовим для ефективного застосування у завданнях оперативного реагування, розслідувань та рятувальних операцій.

## Об'єкт, предмет і мета

**Об'єктом дослідження у кваліфікаційній роботі бакалавра** є процес розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів.

**Предметом дослідження у кваліфікаційній роботі бакалавра** є методи та технології комп'ютерного зору для розпізнавання об'єктів у завданнях розслідувань і пошукових робіт із використанням безпілотних літальних апаратів.

**Метою кваліфікаційної роботи бакалавра** є підвищення ефективності пошукових робіт і розслідувань шляхом розробки методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів.

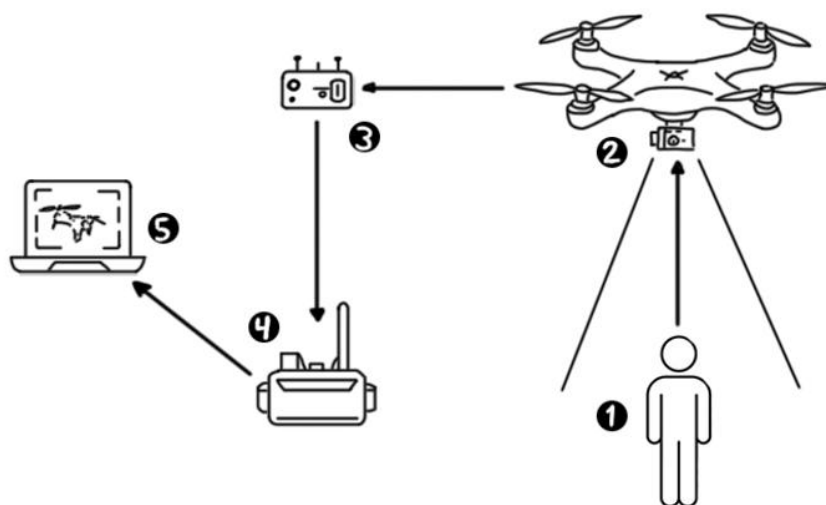
## Задачі роботи

**Задачами кваліфікаційної роботи бакалавра є:**

1. Проведення аналізу актуальності з завдання розробки методу.
2. Наведення досліджень що до наявних підходів що до предметної області цього методу.
3. Розробка методу розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт.
4. Розробка інформаційної систему яка дозволяє реалізувати даний метод.
5. Проведення тестування програмної системи.
6. Проведення експериментальні дослідження ефективності розробленого методу.  
Метод має виконувати наступні основні групи функцій:
  1. Захоплення відеопотоку.
  2. Завантаження моделі.
  3. Виявлення об'єктів.
  4. Візуалізація.
  5. Логування.
  6. Завершення роботи.

## Послідовність передачі відеоданих від БПЛА до комп'ютера для обробки в системі комп'ютерного зору

5

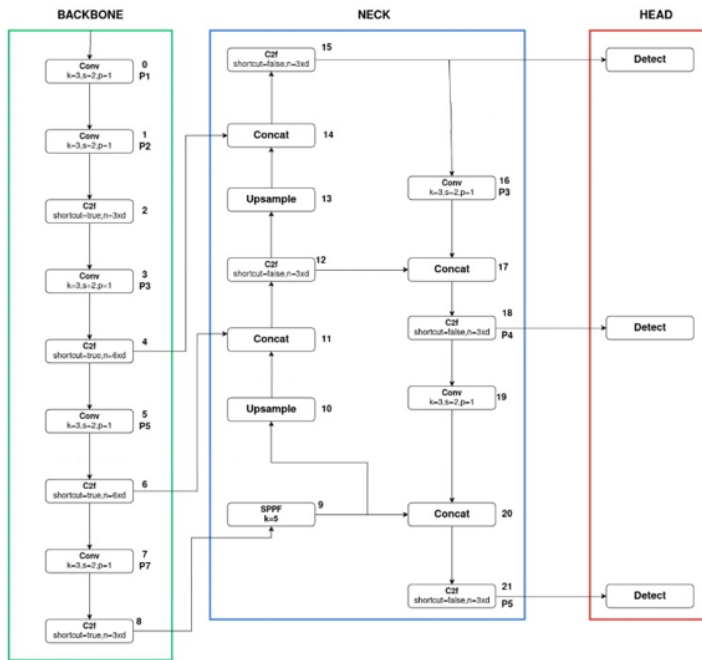


6



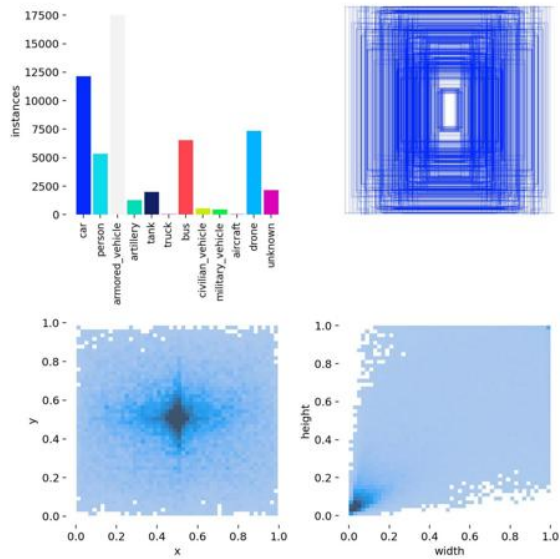
Схема методу  
виявлення об'єктів на  
відеопотоці



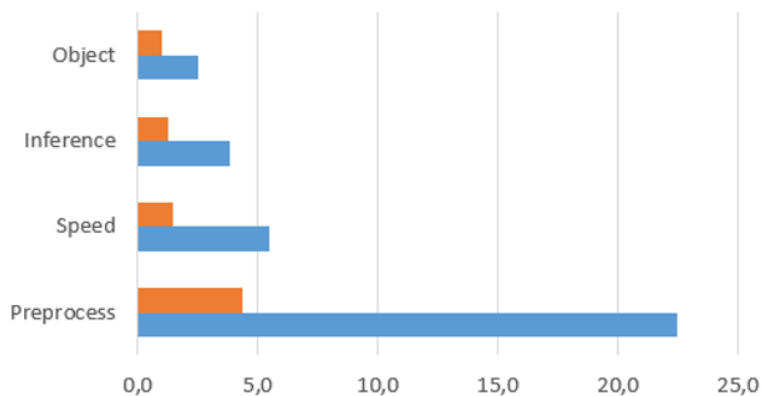


Загальна архітектура  
YOLOv8 та її основні  
частини.

## Комбінована аналітична панель

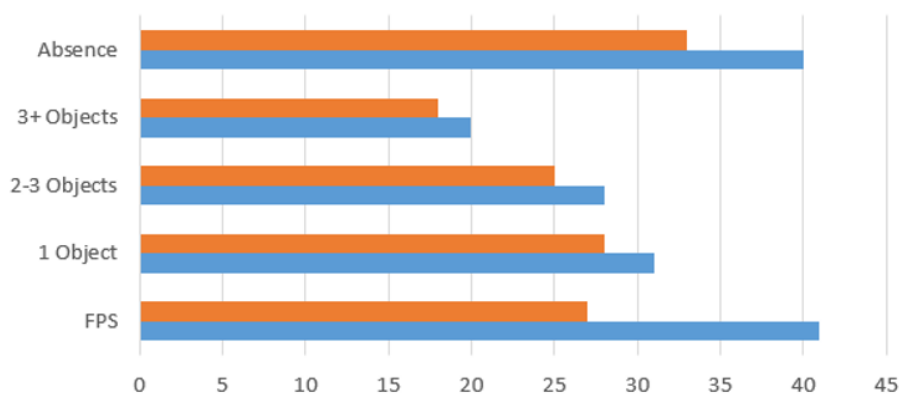


## Динаміка продуктивності методу



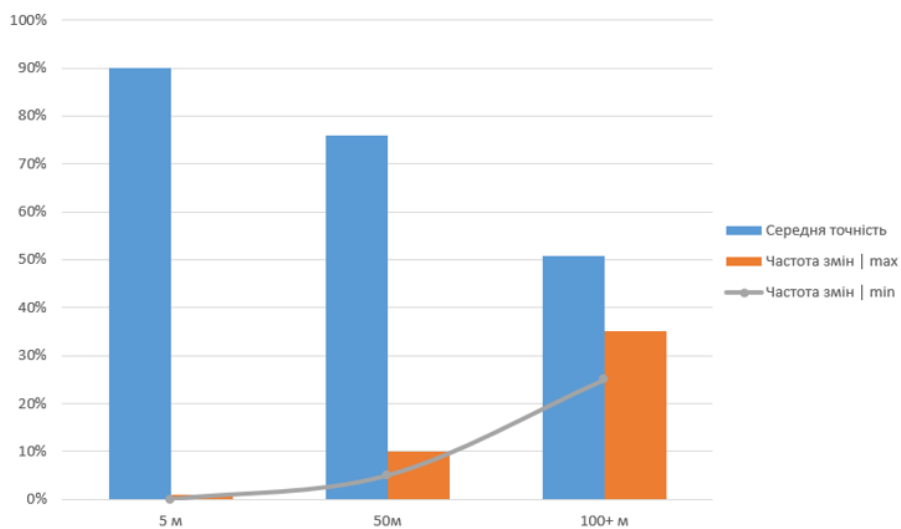
	Preprocess	Speed	Inference	Object
Стандартне відхилення	4,3	1,4	1,2	1
Середнє значення	22,5	5,5	3,9	3

## Залежність FPS від кількості об'єктів у сцені



	FPS	1 Object	2-3 Objects	3+ Objects	Absence
min	27	28	25	18	33
max	41	31	28	20	40

## Показники точності



## Приклад роботи методу



## Приклад роботи методу

15



16

## Висновки

У рамках виконання кваліфікаційної роботи бакалавра було **реалізовано метод** розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт. У ході виконання роботи проведено аналіз предметної області, досліджено сучасні методи комп'ютерного зору, а також вивчено наявні програмні реалізації, зокрема архітектури YOLOv5/v8, які є актуальними для завдань детекції у складних умовах.

Розроблений **метод виконує** захоплення відеопотоку, завантаження моделі YOLOv8, виявлення об'єктів у реальному часі, візуалізацію результатів поверх зображення та логування в консоль. Після завершення обробки забезпечується коректне завершення роботи.

Для створення методу **було використано** Python та бібліотеки PyTorch, Ultralytics YOLO, OpenCV, PyQt5, що забезпечили швидкодію та можливість масштабування. Метод придатний для розслідувань, моніторингу та пошуково-рятувальних завдань.

ДЯКУЮ ЗА УВАГУ!

---

# Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 3.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. **Errors in the documents: 12%**

ID: 244330 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт Added in a DB: 2025-06-09 Authors: Іван ОЛІЙНИК Heads: Едуард МАНЗЮК Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	79385	1179	3168 (4%)	50 (4%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## 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: Іван ОЛІЙНИК

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

Науковий керівник: Едуард МАНЗЮК, д.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:4%

Коефіцієнт подібності 2:1.9%

Мікропробіли: 0

Заміна букв: 3

Інтервали: 0

Білі знаки: 16

Дата створення звіту: 2025-06-09 21:38:37.0

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:

2025-06-10

Дата

експерт

*П. П. Мельничук*

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

Автор студент групи КН-21-2 Іван ОЛІЙНИК

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: д.т.н., проф. каф. комп'ютерних наук Едуард МАНЗЮК

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим 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 детальні 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 доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття 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 Івана ОЛІЙНИКА, 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.

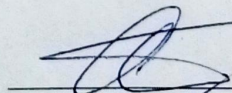
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:

- за системою Anti-Plagiarism: 3%;

- за системою StrikePlagiarism КП1: 4%, КП2: 1,9%.

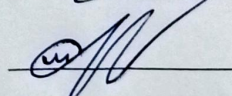
09.06.2025

Завідувач кафедри



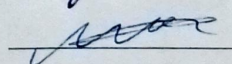
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Едуард МАНЗЮК



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента гр. КН-21-2 Олійник Іван Анатолійович

за темою: Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

1. Актуальність обраної теми

З огляду на активне впровадження БПЛА у цивільних та військових сферах, тема роботи є актуальною та затребуваною

2. Повнота розкриття мети та завдань роботи

Мета роботи чітко визначена, а сам розробка методу для БПЛА. Завдання послідовно виконані, включаючи аналіз, проєктування, реалізацію, дослідження та оцінку ефективності розробленого рішення.

3. Зміст кожного розділу роботи

Перший розділ присвячено аналізу сучасних підходів до розпізнавання об'єктів із борту БПЛА, теоретичним основам обробки відеоданих у реальному часі, огляду існуючого ПЗ, а також формулюванню мети, задач і вимог до розробки методу. У другому описується запропонований метод, його функціональна структура, архітектура нейронної мережі, підготовка даних та використання спеціалізованих програмних компонентів. Третій розглядає засоби реалізації ПЗ, вибір інструментів розробки, структуру та функції програмних модулів, особливості впровадження, а також результати досліджень і аналіз функціональності системи.

4. Оцінка розробленого методу та його практична цінність

Розроблений метод демонструє достатньо високу точність, швидкість та стабільність при малій та великій кількості об'єктів. Практична цінність полягає у можливості оперативного аналізу відеоданих у складних умовах без необхідності доопрацювання апаратного забезпечення.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота відповідає вимогам до оформлення, структурно повна, містить усі обов'язкові розділи, посилання, графічні матеріали та посилання.

6. Недоліки кваліфікаційної роботи бакалавра

Головним недоліком методу є тривале навчання датасетів, що потребує додаткового часу та ресурсів, а також відсутність розробленого графічного інтерфейсу, що ускладнює практичне застосування без технічної підготовки.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Робота заслуговує оцінці "Відмінно"

Рецензент к.т.н., доцент каф. АКТІТРАР  
Корсенька Л.О.

*[Handwritten signature]*



**ВІДГУК НАУКОВОГО КЕРІВНИКА  
на кваліфікаційну роботу бакалавра**

студента сп. гр. КН-21-2 Івана ОЛІЙНИКА

за темою Метод розпізнавання об'єктів у реальному часі для безпілотних літальних апаратів у завданнях розслідувань і пошукових робіт

**1. Актуальність теми**

Тема кваліфікаційної роботи є актуальною з огляду на стрімкий розвиток технологій безпілотних літальних апаратів та зростаючу потребу в інтелектуальних системах комп'ютерного зору для автоматизованої обробки візуальної інформації. Розробка ефективних методів розпізнавання об'єктів у реальному часі є критично важливим завданням для підвищення ефективності пошуково-рятувальних операцій, розслідувань та оперативного реагування на надзвичайні ситуації.

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

Виконана робота повною мірою відповідає стандарту спеціальності 122 – Комп'ютерні науки. Об'єктами дослідження є алгоритми комп'ютерного зору, математичні моделі нейронних мереж, методи обробки відеопотоків та технології машинного навчання. У роботі використано сучасні підходи до розв'язання прикладних задач комп'ютерних наук, зокрема архітектуру YOLOv8, глибоке навчання та технології реального часу.

**3. Професійні та особистісні якості бакалавра**

Під час виконання кваліфікаційної роботи Олійник Іван проявив себе як компетентний дослідник з глибоким розумінням сучасних технологій машинного навчання. Студент демонстрував високий рівень самоорганізації, дотримувався встановлених термінів виконання етапів роботи та виявляв ініціативність у пошуку технічних рішень. Особливо варто відзначити здатність до критичного аналізу існуючих підходів та вміння адаптувати складні алгоритми до практичних завдань.

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

Усі результати, представлені в роботі, отримані студентом самостійно. Олійник Іван особисто здійснював підбір та аналіз наукових джерел, розробляв

архітектуру системи, реалізовував програмні модулі, проводив навчання нейронної мережі та експериментальні дослідження.

#### **5. Ступінь оволодіння методами дослідження**

Студент продемонстрував високий рівень володіння сучасними методами дослідження в галузі комп'ютерного зору та машинного навчання. Успішно застосовував методи глибокого навчання, техніки обробки зображень, алгоритми детекції об'єктів та інструменти візуалізації.

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

Тема роботи розкрита повно та якісно. Проведено ґрунтовний аналіз сучасних підходів до розпізнавання об'єктів, обґрунтовано вибір архітектури YOLOv8, детально описано функціональну структуру методу та особливості його реалізації. Розроблено працездатну систему, проведено комплексне тестування та експериментальні дослідження, які підтверджують ефективність запропонованого підходу.

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

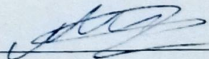
Робота має чітку логічну структуру, матеріал викладено послідовно та аргументовано. Усі розділи логічно пов'язані між собою та підпорядковані досягненню поставленої мети. Використано сучасні наукові джерела, правильно оформлені посилання. Текст написано грамотною українською мовою з дотриманням термінологічних стандартів галузі. Ілюстративний матеріал доповнює текстовий виклад та полегшує сприйняття технічної інформації.

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

Розроблений метод має значний потенціал для практичного застосування в системах безпеки, оборони, пошуково-рятувальних службах та правоохоронних органах. Модульна архітектура системи дозволяє адаптувати її до різних завдань моніторингу та розвідки. Окремі компоненти роботи можуть бути використані для розробки систем відеоаналітики, автоматизованого спостереження та інтелектуального аналізу відеопотоків з безпілотних літальних апаратів.

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

Кваліфікаційна робота бакалавра Олійника Івана виконана на високому рівні, відповідає всім вимогам до робіт даного типу та може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник  д.т.н., проф. каф. КН Едуард МАНЗЮК