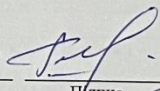


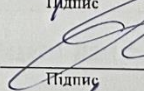
Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук


## КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему Метод виявлення багатороторних БПЛА засобами штучного інтелекту


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

Виконав: студент 2 курсу, група КНм-22-1  С.В. Горшлов  
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: д.т.н., професор, зав. кафедри КН  О.В. Бармак  
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор  О.В. Бармак  
Підпис Ініціали, прізвище

10 грудня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

« 01 » вересня 2023 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

1. Тема кваліфікаційної роботи магістра: «Метод виявлення багатороторних БПЛА засобами штучного інтелекту»

2. Завдання видано студенту Горслову Сергію Вікторовичу

(прізвище, ім'я, по батькові)

3. Керівник роботи завідувач кафедри КН Бармак Олександр Володимирович

(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 15 » серпня 2023 р. № 30

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

Мета кваліфікаційної роботи магістра – розробка методу виявлення багатороторних БПЛА методами штучного інтелекту. Для досягнення поставленої мети визначено наступні задачі: провести аналіз нейромережесевих моделей та існуючих підходів для виявлення БПЛА; розробити метод виявлення багатороторних БПЛА з використанням згорткової нейронної мережі YOLOv8, механізму GAM та Swin Transformer; підготувати набір даних для навчання модифікованої моделі мережі; навчити вдосконалену модель виявляти БПЛА на зображеннях з різними умовами середовища; визначити якість виявлення БПЛА запропонованим методом.

## Реферат

Кваліфікаційна робота магістра присвячена розробці методу виявлення багатороторних БПЛА засобами штучного інтелекту.

**Актуальність теми.** Тема розпізнавання багатороторних безпілотних літальних апаратів (БПЛА) засобами штучного інтелекту в останні роки набула великої актуальності у сучасному науково-технічному середовищі. Зростання популярності БПЛА спостерігається у різноманітних сферах життєдіяльності – від аграрної та геодезійної галузей до використання в мілітарних цілях і системах нагляду. Це зумовлено різноманітністю їх застосувань, здатністю виконувати завдання в недоступних або небезпечних для людей зонах, а також їх ефективністю та доступністю.

У зв'язку з цим, ідентифікація та моніторинг БПЛА стають критично важливими задачами, особливо у контексті забезпечення національної безпеки. Розпізнавання багатороторних БПЛА методами штучного інтелекту відіграє ключову роль у вирішенні цього питання. Штучний інтелект, а конкретно – машинне навчання та нейронні мережі, демонструють значні результати в детектуванні та класифікації об'єктів у реальному часі, навіть у складних умовах.

Актуальність дослідження посилюється високою здатністю сучасних методів штучного інтелекту адаптуватися до динамічних умов реального середовища, вирішувати завдання з великою точністю та ефективністю, що є умовою для покращення систем розпізнавання БПЛА.

**Мета і задачі роботи.** Метою кваліфікаційної роботи магістра є розробка методу виявлення багатороторних БПЛА засобами штучного інтелекту.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз нейромережевих моделей та існуючих підходів для виявлення БПЛА;
- удосконалити модель мережі YOLOv8;
- підготувати набір даних для донавчання моделі;
- провести валідацію запропонованого методу виявлення багатороторних БПЛА за стандартними статистичними показниками.

**Об'єкт дослідження** – процес виявлення багатороторних БПЛА засобами штучного інтелекту.

**Предмет дослідження** – моделі нейронної мережі, методи класифікації ознак для виявлення багатороторних БПЛА.

**Методи дослідження**, застосовані для вирішення поставлених завдань: методи та засоби штучного інтелекту (моделі глибокого навчання, згорткові неймережі), статистичні показники якості моделей глибокого навчання.

**Наукова новизна одержаних результатів.** В результаті проведеної роботи було вдосконалено метод виявлення багатороторних БПЛА засобами штучного інтелекту, який дозволяє інтегрувати модуль високої деталізації для підвищення точності виявлення дрібних повітряних об'єктів. Також було вдосконалено архітектуру неймережі шляхом видалення елементів, що є більш важливими для великих об'єктів, тим самим зменшуючи обчислювальне навантаження та прискорюючи швидкість виявлення БПЛА. Крім того, для кращого виявлення об'єктів різних масштабів та зменшення обчислювальної складності, використано Swin Transformer. Також імплементація модуля уваги GAM у архітектуру мережі значно підвищила здатність до асиміляції ознак, що помітно підвищило відсоток виявлення БПЛА.

**Апробація результатів кваліфікаційної роботи магістра та публікації.** Результати роботи прийняті до публікації у науковому журналі – Метод виявлення багатороторних БПЛА засобами штучного інтелекту / Горєлов С.В., Бармак О.В., Манзюк Е.А // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023. №6 (довідка з редакції у додатках)

**Структура та обсяг роботи.** Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 41 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 106 сторінок, з них 86 сторінок основного тексту та 20 сторінок додатків. У роботі наведено 45 рисунків.

**Ключові слова:** *БПЛА, виявлення об'єктів, YOLOv8, глибоке навчання, GAM, Swin Transformer.*

## Зміст

Перелік скорочень .....	4
Вступ .....	5
РОЗДІЛ 1 .....	8
Огляд підходів до виявлення БПЛА засобами штучного інтелекту .....	8
1.1. Класифікація БПЛА .....	8
1.2 Традиційні методи виявлення БПЛА .....	13
1.3 Аналіз існуючих публікацій виявлення БПЛА засобами ШІ .....	16
1.4 Постановка задачі .....	22
Висновки до розділу 1 .....	23
РОЗДІЛ 2 .....	24
Метод виявлення багатороторних БПЛА засобами ШІ .....	24
2.1 Особливості моделі глибокого навчання YOLOv8 .....	24
2.2 Покращення здатності моделі до виявлення малих цілей .....	28
2.3 Видалення детекційного шару для виявлення великих цілей .....	30
2.4 Трансформери .....	31
2.4.1 Архітектура трансформера зору .....	35
2.4.2 Архітектура Swin Transformer .....	37
2.5 Вдосконалення модуля злиття ознак .....	41
2.6 Статистичні показники для оцінювання якості отриманої моделі глибокого навчання .....	42
Висновки до розділу 2 .....	44
РОЗДІЛ 3 .....	45
Проектування структури інформаційної системи для експериментального дослідження запропонованого методу .....	45
3.1 Засіб розробки та тестування моделей машинного та глибокого навчання Python .....	45
3.2 Пакет Ultralytics для використання YOLOv8 .....	46
3.3 Платформа машинного навчання PyTorch .....	50
3.4 Функціональність по донавчанню мережі на власному наборі даних .....	51

3.5 Функціональність для вдосконалення архітектури мережі YOLOv8 .....	52
Висновки до розділу 3.....	54
РОЗДІЛ 4 .....	56
Експериментальні результати виявлення багатороторних БПЛА запропонованим методом.....	56
4.1 Опис застосування для проведення експериментів .....	56
4.2 Опис набору даних для тестування та параметрів навчання .....	61
4.3 Власний набір даних для донавчання нейромережі .....	64
4.4 Вдосконалення архітектури мережі YOLOv8 .....	66
4.5 Порівняння отриманих результатів тестування на наборі даних UAV TIB-Net.....	68
4.6 Порівняння отриманих результатів тестування на власому наборі даних.....	73
Висновки до розділу 4.....	79
ВИСНОВКИ .....	81
Перелік посилань.....	83
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БПЛА	Безпілотні літальні апарати
GAM	Global attention mechanism
MSA	Mechanism of Self-Attention
RNN	Recurrent neural networks
CNN	Convolutional neural network
SVM	Support vector machines
CPU	Central processing unit
GPU	Graphics processing unit
FPS	Frames per second

## Вступ

**Актуальність теми.** Тема розпізнавання багатороторних безпілотних літальних апаратів (БПЛА) засобами штучного інтелекту в останні роки набула великої актуальності у сучасному науково-технічному середовищі. Зростання популярності БПЛА спостерігається у різноманітних сферах життєдіяльності – від аграрної та геодезійної галузей до використання в мілітарних цілях і системах нагляду. Це зумовлено різноманітністю їх застосувань, здатністю виконувати завдання в недоступних або небезпечних для людей зонах, а також їх ефективністю та доступністю.

У зв'язку з цим, ідентифікація та моніторинг БПЛА стають критично важливими задачами, особливо у контексті забезпечення національної безпеки. Розпізнавання багатороторних БПЛА методами штучного інтелекту відіграє ключову роль у вирішенні цього питання. Штучний інтелект, а конкретно – машинне навчання та нейронні мережі, демонструють значні результати в детектуванні та класифікації об'єктів у реальному часі, навіть у складних умовах.

Актуальність дослідження посилюється високою здатністю сучасних методів штучного інтелекту адаптуватися до динамічних умов реального середовища, вирішувати завдання з великою точністю та ефективністю, що є умовою для покращення систем розпізнавання БПЛА.

**Мета і задачі роботи.** Метою кваліфікаційної роботи магістра є розробка методу виявлення багатороторних БПЛА засобами штучного інтелекту.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз нейромережових моделей та існуючих підходів для виявлення БПЛА на зображеннях;
- розробити метод виявлення багатороторних БПЛА з використанням згорткової нейронної мережі YOLOv8, модуля уваги GAM та Swin Transformer;
- підготувати набір даних для донавчання мережі;

– провести валідацію та аналіз отриманих експериментальних результатів запропонованого методу виявлення БПЛА за стандартними статистичними показниками.

**Об’єкт дослідження** – процес виявлення багатороторних БПЛА засобами штучного інтелекту.

**Предмет дослідження** – моделі нейронної мережі, методи класифікації ознак для виявлення багатороторних БПЛА.

**Методи дослідження**, застосовані для вирішення поставлених завдань: методи та засоби штучного інтелекту (моделі глибокого навчання, згорткові нейномережі), статистичні показники якості моделей глибокого навчання.

**Наукова новизна одержаних результатів.** В результаті проведеної роботи було вдосконалено метод виявлення багатороторних БПЛА засобами штучного інтелекту, який дозволяє інтегрувати модуль високої деталізації для підвищення точності виявлення дрібних повітряних об’єктів. Також було вдосконалено архітектуру нейромережі шляхом видалення елементів, що є більш важливими для великих об’єктів, тим самим зменшуючи обчислювальне навантаження та прискорюючи швидкість виявлення БПЛА. Крім того, для кращого виявлення об’єктів різних масштабів та зменшення обчислювальної складності, використано Swin Transformer. Також імплементація модуля уваги GAM у архітектуру мережі значно підвищила здатність до асиміляції ознак, що помітно підвищило відсоток виявлення БПЛА.

**Апробація результатів кваліфікаційної роботи магістра та публікації.** Результати роботи прийняті до публікації у науковому журналі – Метод виявлення багатороторних БПЛА засобами штучного інтелекту / Горелов С.В., Бармак О.В., Манзюк Е.А // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023. №6 (довідка з редакції про прийом публікації у додатках).

**Структура та обсяг роботи.** Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 41 найменування та 3 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 106 сторінок, з них 86 сторінок основного тексту та 20 сторінок додатків. У роботі наведено 45 рисунків.

## РОЗДІЛ 1

### Огляд підходів до виявлення БПЛА засобами штучного інтелекту

#### 1.1. Класифікація БПЛА

Безпілотний літальний апарат (БПЛА) є автономною або дистанційно керованою літальною системою, оснащеною датчиками, цілевказівниками, наступальним озброєнням або електронними засобами для створення перешкод або нейтралізації ворожих цілей. Відсутність екіпажу та потреби у системах життєзабезпечення дозволяють БПЛА забезпечувати підвищену дальність та витривалість порівняно з маневровими літальними апаратами, управління якими вимагає участі пілота [5].

Роботи зі створення безпілотних літальних апаратів почалися давно, ще в роки Першої світової війни [1]. У 1930-х роках з'явилися перші Дистанційно пілотовані повітряні цілі. Під час Другої світової війни з'явився перший безпілотний штурмовик – німецька пускова установка V-1. Тоді штурмовики дальньої дії були класифіковані як крилаті ракети і називалися безпілотними літальними апаратами. Наприкінці 1950-х років з'явилися безпілотні літальні апарати. 70-ті роки 20 століття сходять до досліджень і розробок в області бойових (ударних) технологій безпілотні літальні апарати, а також безпілотні літальні апарати з великими висотами і періодами польоту, призначені для використання в складі довготривалих комплексів спостереження і розвідки і нанесення ударів. У 1970-і і 1980-і роки цією темою займалося конструкторське бюро А.Н. Туполєва, П.А. Сухого, В.М. Мясіщева, Н.І. Камова, А.С. Яковлєва. Безпілотні літальні апарати все ширше використовуються в інших видах діяльності людини. У сільському господарстві, наприклад, безпілотники з GPS-навігацією використовуються для запилення рослин на полях [2]. У той же час досягається значна економія хімікатів і більш ретельна обробка посівів в порівнянні з пілотованою авіацією. Вони використовуються для доставки медикаментів і гуманітарних вантажів у важкодоступні райони і можуть

використовуватися для перевірки трубопровідних ліній електропередачі. Дрони, інша назва безпілотних літальних апаратів, знайшли своє використання у Національній службі з надзвичайних ситуацій, вони використовуються для контролю небезпечних об'єктів (безпека і охорона навколишнього середовища) і багато іншого. Зокрема, мультикоптерні безпілотні літальні апарати можуть використовуватися як несуча платформа для метеорологічних вимірювальних комплексів [3]. Вони мають переваги по відношенню до безпілотних літальних апаратів літального типу, які не забезпечують достатнього просторового і часового дозволу вимірювань через їх високу швидкість переміщення в атмосфері, низьку чутливість і нездатність вивчати турбулентні процеси. В даний час доступні і відносно недорогі мультикоптери пропонують підйомники з корисним навантаженням 30-40 кг на висоту 2-4 км з часом польоту 30-40 хвилин. У сучасному безпілотнику бортова система навігації та управління може забезпечити:

- політ по заданому маршруту (маршрут задається координатами і висотою поворотної точки маршруту);
- зміну призначення маршруту або повернення в вихідну точку по команді з наземного пункту управління;
- обліт зазначеної точки;
- стабілізацію кута орієнтації БПЛА, підтримку заданої висоти і швидкості польоту;
- збір і передачу телеметричної інформації і параметрів польоту.

Все це забезпечує більшу мобільність і ефективність вимірювань при низьких витратах при експлуатації безпілотних літальних апаратів. Основні переваги технології вимірювання параметрів приграничного шару атмосфери (APS) з використанням безпілотного літального апарату мультикоптерного типу:

- пряме контактне вимірювання метеорологічних і турбулентних характеристик APS;

- можливість оперативного короткочасного вимірювання характеристик APS в заданих точках атмосфери (висота від 0 до 4000 метрів);
- хороша завадостійкість;
- можливість вимірювання вертикального профілю погоди, турбулентності і характеристик навколишнього середовища точки доступу за допомогою керованого підйому (спуску) безпілота відповідно до заданої програми;
- мала вага і габарити пристрою;
- низька ціна.

Основні недоліки технології вимірювання параметрів APS з використанням мультикоптерного безпілотної літального апарату:

- мала тривалість вимірювання (20-40 хвилин);
- відносно низька вантажопідйомність несучої платформи (вимагає створення спеціалізованих вимірювальних систем з малою вагою, невеликими габаритами і низьким енергоспоживанням).

Багато невеликих безпілотної літальних апаратів стають доступними широкому користувачеві, а їх обладнання настільки складне, що включає в себе фото і відеокамери, автопілот і навігаційні системи, що дуже спрощує управління такими дронами.

В залежності від рівня автономності та способу управління, БПЛА поділяються на такі категорії:

- 1) дистанційно-керовані БПЛА;
- 2) некеровані БПЛА;
- 3) автоматичні БПЛА.

Що стосується конструктивних особливостей, то БПЛА можна класифікувати за типом їхніх крил на:

- 1) багатороторні БПЛА;
- 2) безпілотики з фіксованим крилом;

- 3) однороторні вертолітні БПЛА;
- 4) гібридні БПЛА VTOL з фіксованим крилом.

Ці категорії відображають різноманітність конструкцій та можливостей БПЛА, що дозволяє їм виконувати широкий спектр завдань, від розвідки до виконання цільових ударів.

Хоча в академічних джерелах часто зустрічається термін БПЛА, існує кілька інших назв та скорочень для літальних апаратів, які функціонують без присутності людини на борту. Ці терміни включають: безпілотну літальну систему (UAS), віддалено пілотований літальний апарат (RPA) або транспортний засіб (RPV), та віддалено керований літальний апарат (ROA). Також існує термін "безлюдний літальний апарат". Хоча технічно ці терміни мають свої відмінності [6], іноді вони можуть викликати певну семантичну плутанину. Наприклад, термін БПЛА часто асоціюється з військовим застосуванням, тоді як UAS зазвичай використовується для опису апаратів, що застосовуються в цивільних цілях, як аматорами, так і професіоналами.

У роботі розглядаються багатороторні БПЛА так, як вони є найбільш широко використовуваним типом безпілотників для рекреаційного та професійного використання. На сьогоднішній день на території України у військових операціях широко використовується саме цей тип БПЛА. Ці безпілотники військові підрозділи використовують для ведення розвідки та виявлення цілей для вогневого враження [7].

Поширення невеликих безпілотників, крім їх безсумнівних позитивних сторін, створило безліч проблем, пов'язаних з неналежною поведінкою деяких власників БПЛА, несанкціонованим моніторингом об'єктів і територій національного значення, зростаючими порушеннями конфіденційності, появою можливості використання безпілотників. Незалежно від області застосування, повноцінна реалізація завдань БПЛА може включати в себе такі підзадачі, як виявлення цілей, локалізація та ідентифікація, відстеження і наведення на ціль.

Повітряна розвідка вважається однією з найнебезпечніших бойових завдань. Виявлені цілі, інша сторона намагається приховати і захистити їх за допомогою потужних об'єктів і військової протиповітряної оборони. Повітряна розвідка особливо небезпечна на початку бойових дій, коли протиповітряна оборона противника ще не пригнічена і відсутня перевага у повітряному просторі. Тому особливо актуальне застосування комплексів на базі безпілотних літальних апаратів для повітряної розвідки [4].

У більшості перелічених випадків дуже актуальним стає завдання виявлення БПЛА у повітрі.

Отже, можна зробити висновок, що застосування БПЛА має значний прогрес у сфері авіаційних та технологічних розробок, використовуючись від військових до цивільних застосувань. Започатковані як засоби ведення військових дій, ці системи з часом трансформувалися у багатофункціональні інструменти для вирішення широкого спектру задач, що охоплюють розвідку, моніторинг довкілля, сільське господарство та навіть доставку. Попри численні переваги, такі як відсутність необхідності у пілота на борту та здатність виконувати тривалі і складні місії, існують значні виклики, зокрема у сферах безпеки, конфіденційності та регуляції. Сукупність цих факторів робить БПЛА важливим елементом сучасного технологічного ландшафту, що вимагає балансу між інноваційним потенціалом та врахуванням етичних та правових аспектів їх застосування.

Отже, було проведено детальний аналіз та огляд історії, розвитку, застосування та перспектив БПЛА. Розглянуто еволюцію БПЛА від простих ранніх моделей до складних сучасних систем, їхню роль у військових та цивільних операціях, а також перспективи використання в різних галузях. Особливу увагу приділено технічним характеристикам, включаючи навігаційні та управлінські системи, а також викладено потенційні можливості та виклики, пов'язані з БПЛА. Значна частина розділу присвячена аналізу різних типів БПЛА, їхніх конструктивних особливостей та сфер застосування. Таким чином, розділ надає

всебічне розуміння ролі та впливу БПЛА у сучасному світі, підкреслюючи їх значення як у військовій, так і в цивільній сфері.

## 1.2 Традиційні методи виявлення БПЛА

Інформація для виявлення і подальшого визначення напрямку руху БПЛА надається за допомогою спеціальних засобів, що фіксують відбиття та випромінювання енергії у всіх діапазонах спектра електромагнітних та акустичних хвиль.

Кожен матеріальний об'єкт, включаючи безпілотний літальний апарат, має демаскуючу особливість, яка виділяє його в навколишньому середовищі, роблячи його видимим для спостереження. В даний час ступінь видимості визначається радіочастотами, інфрачервоним випромінюванням (ІЧ) і значенням його сигнатури у видимому діапазоні спектра, а також акустичної сигнатурою. Сучасні легкі дрони володіють відмінними рисами, що можуть заважати традиційній пеленгації. Вони можуть бути виготовлені з композитних матеріалів і пластика зі спеціальним забарвленням і особливим поєднанням шарів. Вони мають невеликий бензиновий або електричний двигун, який майже не виділяє тепла і працює практично безшумно [8].

Значна різноманітність можливих варіантів побудови та використання інструментів моніторингу в кожному з цих діапазонів ускладнює оцінку їх ефективності.

Методи виявлення БПЛА можна класифікувати наступним чином:

- аудіодетекція;
- візуальне виявлення;
- теплове виявлення;
- радіолокаційне виявлення;
- радіочастотне (RF) виявлення;
- виявлення через Wi-Fi.

Аудіодетекція базується на зовнішніх звуках, які при роботі видає БПЛА. Ці звуки можуть бути зафіксовані та перевірені згідно з існуючою базою даних відомих акустичних профілів БПЛА. Проте, в галасливих умовах, таких як міські райони, цей метод може бути неефективним. Окрім того, БПЛА вищого класу часто оснащуються спеціальними гвинтами та двигунами, що змінюють їх акустичний відбиток[9].

Візуальне виявлення використовує камери для визначення положення та руху повітряних об'єктів, намагаючись розрізнити БПЛА від птахів за розміром, траєкторією польоту та стилем руху [11]. Пластиковий корпус БПЛА не проводить тепло, а двигуни виробляють менше тепла, ніж може здатися. Тим не менш, літєві акумулятори, які живлять більшість БПЛА, генерують достатньо тепла для виявлення інфрачервоними камерами. Ці камери ефективні за різниці температур і здатні "бачити" у повній темряві, що робить їх ідеальними для нічного використання або при виконанні непомітних місій. Однак, деякі характеристики польоту БПЛА схожі на ті, що виконуються птахами, що може ускладнювати виявлення.

Теплове виявлення базується на фіксації теплової сигнатури БПЛА, яка як було сказано раніше, зазвичай мінімальна через використання пластикових матеріалів у їхній конструкції.

Радіо (RF) виявлення полягає у моніторингу частот 2,4 та 5,8 ГГц, які часто використовуються БПЛА для передачі даних[10].

Wi-Fi виявлення можливе завдяки трансляції ідентифікаційних SSID і MAC-адрес багатьма комерційними БПЛА. На Рисунку 1.1 ілюструється сценарій зловмисного використання дрону, який керується з дистанції та літає поблизу вразливих об'єктів. Для ефективного захисту система виявлення повинна бути розміщена в місці, яке потребує захисту. Ця система повинна виявляти висхідні сигнали (керування) та низхідні (відео та телеметрія) та передавати інформацію про присутність та місцезнаходження БПЛА на центральний сервер. Дальність

виявлення залежить від таких факторів, як ланцюг RF-підсилення, апаратні компоненти та програмні процеси, що використовуються у системі.

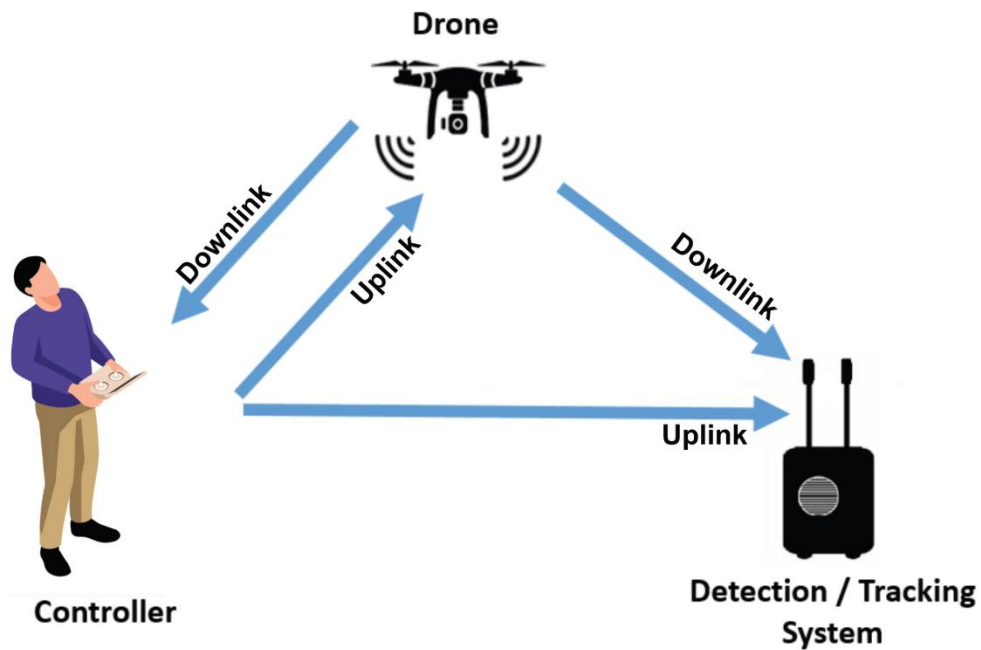


Рисунок 1.1 – Сценарій виявлення та відстеження БПЛА, що використовують Wi-Fi: низхідна лінія - Downlink (відео та телеметрія)/висхідна лінія - Uplink (контроль) [10]

Радарне виявлення забезпечує ефективне виявлення БПЛА на значній відстані і може бути інтегроване з іншими технологіями для забезпечення всебічного покриття. Радіолокаційна система складається з передавача, що випромінює радіохвилі. Ці хвилі відбиваються або розсіюються від об'єктів та повертаються до приймача, де алгоритми перетворюють їх у візуальні зображення, надаючи інформацію про форму, розмір та щільність об'єкта. Використання радарів у аеропортах включає комбінацію далекодійних та короткодійних радарів для управління повітряним рухом. Однак, через невеликі розміри БПЛА, потрібен спеціалізований підхід, а саме, використання радарів X-діапазону з коротшими довжинами хвиль (8,0 - 12,0 ГГц) [12].

Імпульсно-доплерівські радіолокаційні системи аналізують спотворення радіохвиль для оцінки відстані, швидкості та характеристик об'єкта. Однак основна

частина БПЛА виготовлена з пластику, який не відображає радіохвилі, і тільки металеві частини, такі як камери, акумулятори та двигуни, можуть відображати радіолокаційні сигнали [13].

Різні характеристики які властиві БПЛА, такі як радіочастотна, інфрачервона, видима та акустична, вимагають застосування різних методів виявлення. Важливо відзначити, що сучасні БПЛА, зроблені з композитних матеріалів та пластику, це ускладнює їх виявлення. Кожен із методів, такі як аудіодетекція, візуальне, теплове, радіолокаційне та радіочастотне виявлення, має свої переваги та обмеження, і їх ефективність може варіюватися в залежності від умов середовища та характеристик БПЛА. Наприклад, аудіодетекція може бути неефективною у галасливих умовах, а візуальне виявлення може бути ускладненим схожістю характеристик польоту БПЛА та птахів. Теплове виявлення ефективне в нічний час, але мінімальні теплові сигнатури БПЛА ускладнюють цей процес. Радіолокаційне виявлення, незважаючи на свою здатність забезпечувати ефективне виявлення на великих відстанях, також зіткнулося з викликами через невеликі розміри та композитну природу БПЛА. У підсумку, цей розділ підкреслює, що для ефективної пеленгації та виявлення БПЛА потрібен багатогранний підхід, що поєднує кілька методів для досягнення оптимальних результатів.

### **1.3 Аналіз існуючих публікацій виявлення БПЛА засобами ШІ**

Протягом останніх років відзначається стрімкий розвиток технологій обробки візуальних даних, зумовлений підвищенням ефективності обчислювальних систем. Це сприяло значному прогресу в сфері глибокого навчання та використанні великих даних для обробки зображень. Особлива увага дослідників приділяється виявленню об'єктів з використанням методів комп'ютерного зору. За тривалий час досліджень відбулась еволюція методів від ручних, де аналітик або дослідник вручну визначає та обирає конкретні характеристики з датасету, які важливі для подальшого аналізу або класифікації, а також робить виокремлення візуальних елементів, таких

як колір, текстура, форма, контури або розміри об'єктів [14,15,16], до використання глибокого навчання для підвищення точності розпізнавання у візуальному виявленні об'єктів. Це включає застосування згорткових обчислень для виявлення об'єктів, мережі які використовують такий метод обчислень називають –згортковими нейронними мережами (convolutional neural network, CNN, ConvNet).

Згорткові нейронні мережі, введені Яном Лекуном у 1988 році [17], представляють собою унікальну архітектуру штучних нейронних мереж, яка знайшла широке застосування у класифікації зображень. Так, Facebook застосовує ці мережі для автоматизації маркування фотографій, Amazon використовує їх для створення рекомендацій продуктів, а Google - для удосконалення пошуку фотографій.

Основна мета класифікації зображень - визначити, до якої категорії вони належать. Людина може легко справлятися з цим завданням, на відміну від комп'ютера, для якого кожне зображення є лише набором значень пікселів. Тому важливо правильно обробити ці дані, що і робить конволюційна нейронна мережа через кілька ключових шарів:

- *вхідний шар*, який отримує зображення і при необхідності змінює їх розмір для подальшої передачі в мережу, де відбувається розпізнавання об'єктів;
- *згорткові шари*, що діють як фільтри, виділяючи важливі елементи зображення і обчислюючи міру відповідності під час тестування;
- *агрегувальні шари*, що збирають виходи груп нейронів у єдиний нейрон наступного шару, зменшуючи просторові розміри даних, кількість параметрів і обсяг обчислень, тим самим контролюючи перенавчання;
- *шар зрізаних лінійних вузлів*, який перетворює будь-яке від'ємне число з попередніх шарів у нуль, забезпечуючи стабільність в мережі, уникаючи екстремальних значень;
- *повноз'єднаний шар*, який класифікує зображення, визначаючи його категорію після високорівневої фільтрації.

Основним компонентом згорткової нейронної мережі є згортковий шар, який складається з ряду фільтрів. Ці фільтри працюють, пересуваючись по вхідним даним і вчаться в процесі прямого проходу, обчислюючи скалярний добуток між даними фільтру та вхідними даними, представленими значеннями пікселів, формуючи двовимірну карту активації для кожного фільтру.

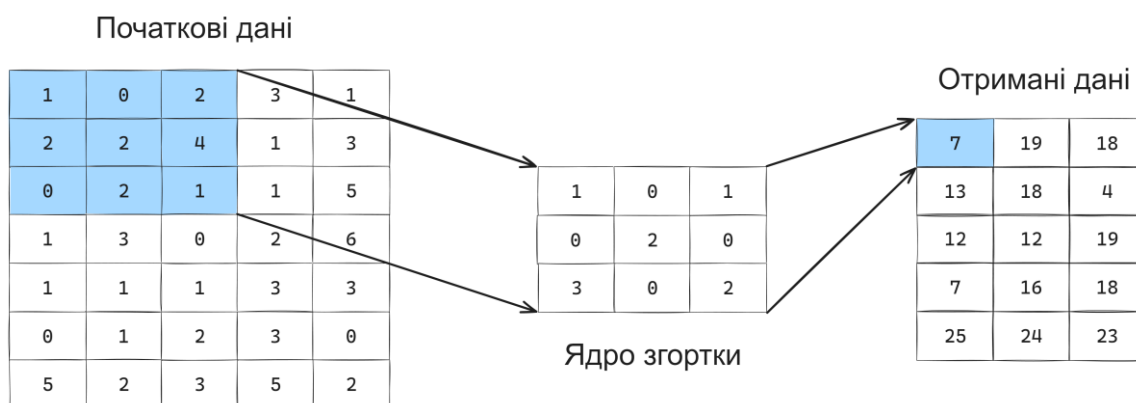


Рисунок 1.2 – Приклад роботи фільтру згорткового шару

Порівняно із цими традиційними методами, виявлення об'єктів за допомогою візуальних сенсорів, зокрема камер, що записують групові відео та зображення, пропонує більш інтуїтивне виявлення та розпізнавання інформації про об'єкти або групи об'єктів. Цей метод пропонує переваги, такі як реальний час та динамічний запис послідовних зображень цілей, кращу швидкість виявлення та стійкість до низьковисотних перешкод [18].

Виявлення об'єктів є ключовим напрямком у галузі комп'ютерного зору, що лежить в основі багатьох складних візуальних завдань. Цей напрямок активно застосовується у промисловості, сільському господарстві та інших сферах [19,20]. З 2015 року відбулися значні прориви у техніках виявлення об'єктів, основаних на глибокому навчанні, з такими відомими алгоритмами, як Faster R-CNN [21], SSD [22] та серія YOLO [23], які значно підвищили точність виявлення об'єктів.

Застосування серії YOLO в області виявлення БПЛА описано в дослідженнях [24,25,26]. Наприклад, у [24] розширення моделі YOLO включає модуль уваги (Attention Module) для підсилення здатності вилучати характеристики.

Модулі уваги використовуються для того, щоб змусити CNN навчатися та фокусуватися більше на важливій інформації, а не на некорисній фоновій інформації. У випадку виявлення об'єктів, корисною інформацією є об'єкти або цільовий клас кадру, який потрібно класифікувати та локалізувати на зображенні.

Модуль уваги складається з простого 2D-конволюційного шару, multilayer perceptron (MLP) та сигмоїдної функції для створення маски вхідної карти ознак.

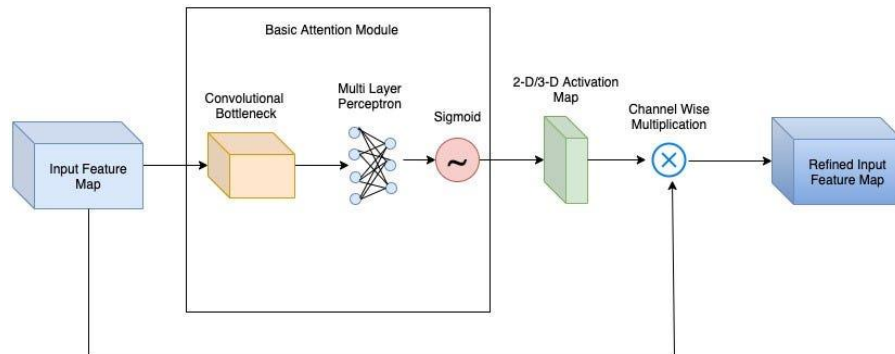


Рисунок 1.3 – Базова структура модуля уваги [28]

В [25] YOLOv3 використовується для виявлення невеликих цілей безпілотних літальних апаратів на складних міських фонах. У [26] MobileNetv2 замінює CSPDarknet53 в YOLOv4, зменшуючи обсяг моделі та спрощуючи обчислювальні процеси.

MobileNets є ефективними нейронними мережами, які засновані на упрощеній архітектурі, використовуючи роздільні згортки по глибині для створення менш об'ємних глибоких мереж. Зменшення розмірів цих мереж досягається за допомогою спеціальних гіперпараметрів.

У дослідженні [28] описується використання MobileNet для системи моніторингу споживання їжі у реальному часі. Ця система вимагає 762 кБайт флеш-пам'яті та 500 кБайт оперативної пам'яті для реалізації на мікроконтролері STM32H7 і здатна обробляти зображення за 300 мс, досягаючи середньої точності розпізнавання 0,82 на тестових зображеннях та 0,74 у реальному часі. Інше дослідження [29] пропонує використання MobileNet у безпілотних автомобілях для

аналізу зображень з камер, спрямованих на визначення дорожніх смуг та прогнозування поворотів.

Використання MobileNets значно розширює функціональність вбудованих систем на мікроконтролерах, надаючи їм можливість ефективного розпізнавання зображень навіть за обмеження ресурсів. Такий підхід демонструє важливість MobileNets у сучасних застосуваннях комп'ютерного зору та їхню здатність адаптуватися до різних умов використання.

Модель MobileNet використовує унікальний підхід до згорток, який включає факторизацію стандартної згортки на дві частини: глибинну згортку та поточкову згортку  $1 \times 1$ . Глибинна згортка призначена для застосування одного фільтра до кожного вхідного каналу окремо, а поточкова згортка  $1 \times 1$  використовується для комбінування вихідних даних після глибинної згортки.

У традиційній згортці один крок обробляє вхідні дані, застосовуючи фільтрування та комбінування для створення нового набору вихідних даних. Відмінно від цього, глибинна роздільна згортка розбиває цей процес на два окремі шари: один для фільтрації та інший для комбінування. Таке розділення дозволяє значно зменшити обчислювальні витрати та загальний розмір моделі[30], роблячи MobileNet особливо ефективним для використання в обладнанні з обмеженими обчислювальними ресурсами.

На Рисунку 1.4 демонструється процес розбиття стандартної згортки на дві складові: згортка по глибині та поточкову згортку  $1 \times 1$ . Стандартний згортковий шар отримує вхідні дані у формі карти функцій  $F$  з розмірами  $D_F \times D_F \times M$  і перетворює їх у вихідну карту функцій  $G$  з розмірами  $D_G \times D_G \times N$ , де  $D_F$  є просторовою шириною та висотою вхідної карти функцій,  $M$  - кількість вхідних каналів,  $D_G$  - просторова ширина та висота вихідної карти функцій, а  $N$  - кількість вихідних каналів.

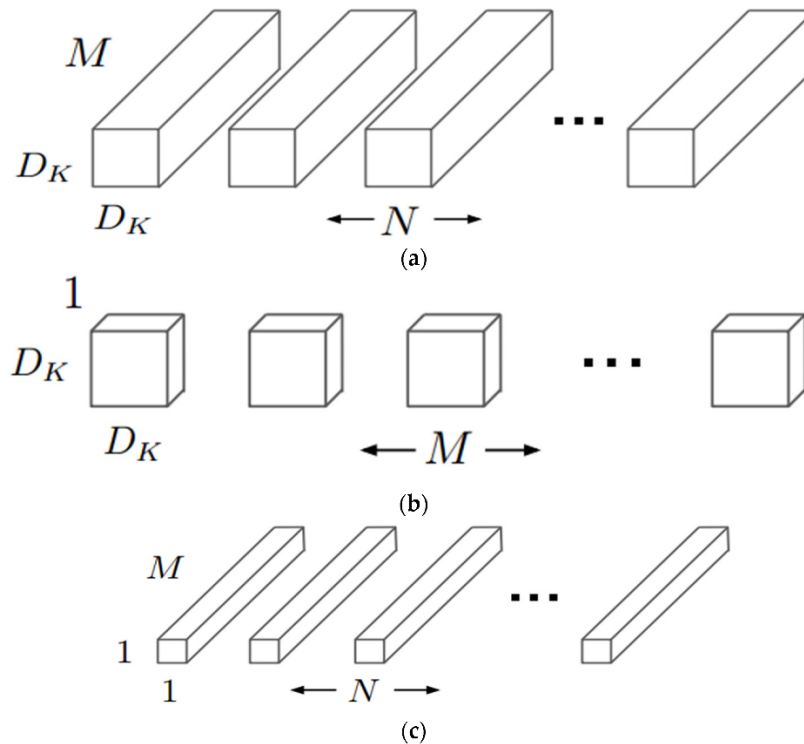


Рисунок 1.4 – Етапи перетворення згорткових фільтрів: а) стандартна згортка, б) згортка по глибині, в) поточкова згортка [30]

Стандартний згортковий шар використовує ядро згортки  $K$  з розмірами  $D_K \times D_K \times M \times N$ , де  $D_K$  – просторовий розмір ядра, зазвичай квадратного,  $M$  – кількість вхідних каналів, а  $N$  – кількість вихідних каналів.

Цей процес розкладання дозволяє реалізувати оптимізовану версію стандартної згортки, зменшуючи обчислювальні витрати та розмір моделі без істотного впливу на продуктивність.

Основною відмінністю MobileNet V2 від попередньої версії є введення нового модуля в архітектуру: інвертований залишковий блок із лінійними вузькими місцями (bottleneck layers). Цей блок спочатку розширює стиснені вхідні дані, збільшуючи їх розмір, а потім застосовує глибинну згортку для фільтрації. Після цього оброблені дані проєктуються назад у зображення за допомогою лінійної згортки[31].

Цей підхід дозволяє зменшити розмір вхідних даних, що важливо для ефективної роботи на пристроях з обмеженими обчислювальними ресурсами, таких як мобільні телефони та інші портативні пристрої. Інвертовані залишкові блоки з

вузькими місцями забезпечують збереження ефективності обчислень, одночасно підтримуючи високий рівень продуктивності. Ці блоки можуть бути легко інтегровані в стандартні операції будь-якого сучасного фреймворку для глибинного навчання.

Завдяки цим інноваціям, MobileNet V2 став особливо привабливим для мобільних додатків, де критично важливо зменшити обсяг пам'яті, потрібний для зберігання та обробки даних, не втрачаючи при цьому точності розпізнавання.

Проаналізувавши дослідження, присвячені розвитку технологій обробки візуальних даних, можемо зробити висновок, що велика увага приділяється виявленню безпілотних літальних апаратів за допомогою згорткових нейронних мереж. Розвиток цих технологій демонструє значний перехід від ручних методів обробки зображень до застосування алгоритмів глибокого навчання, які підвищують точність і швидкість виявлення об'єктів. Особливо ефективними в цьому аспекті виявилися алгоритми, такі як серія YOLO, що забезпечують високу точність виявлення БПЛА на складних фонах. Такий аналіз підкреслює важливість та потенціал CNN у візуальному розпізнаванні та моніторингу БПЛА, відкриваючи нові перспективи для покращення систем безпеки та спостереження.

#### **1.4 Постановка задачі**

Метою кваліфікаційної роботи магістра є розробка методу виявлення багатороторних БПЛА засобами штучного інтелекту. Для виконання поставленого завдання потрібно:

- провести аналіз нейромережевих моделей та існуючих підходів для виявлення багатороторних БПЛА;
- розробити метод виявлення багатороторних БПЛА у полі зору камери, використовуючи нейромережевий засіб виявлення об'єктів (YOLOv8);
- підготувати набір даних для донавчання нейронної мережі YOLOv8;

- провести процес навчання нейромережі без використання попередньо навченої моделі для виявлення малих БПЛА у повітрі;
- порівняти отриману модель з початковою та різними її модифікаціями використовуючи відповідні статистичні показники.

Вдалим виконанням завдання можна вважати отримання меншої за розміром моделі та покращеними параметрами розпізнавання дрібних об'єктів.

### **Висновки до розділу 1**

У процесі наведеного аналізу була визначена чітка мета дослідження та сформульовані ключові завдання. В ході роботи була підкреслена актуальність вивчення та розробки методів виявлення та ідентифікації БПЛА, особливо в контексті зростаючих вимог до систем повітряного спостереження та безпеки. Було проведено детальний огляд існуючих наукових робіт, що висвітлюють сучасні підходи та технології в області оптичного розпізнавання БПЛА, що підтверджує значущість та практичну цінність досліджуваної теми.

Зроблена постановка задачі.

## РОЗДІЛ 2

### Метод виявлення багатороторних БПЛА засобами ШІ

Реалізація ефективного рішення для розв'язування поставленого завдання базується на модифікації існуючих рішень або поєднанні декількох рішень з метою отримання ефективного засобу розв'язку поставленої задачі. Об'єднуючи ідеї поліпшення мережі, які запропоновані у проаналізованій літературі щодо серії YOLO, у цій роботі буде проведено вдосконалення моделі YOLOv8s, щоб вона була придатна для виявлення маленьких об'єктів БПЛА. Виходячи з цього, основна суть запропонованого методу полягає у використанні вдосконаленої моделі глибокого навчання. Вдосконалення полягає у наступному.

- *Етап 1: Покращення здатності моделі до виявлення малих цілей.* Для цього пропонується інтегрувати шар високої роздільної здатності в частину head та вилучити шари виявлення та комбінування ознак для прогнозування великих цілей.
- *Етап 2: Вдосконалення backbone.* Це буде зроблено шляхом інтеграції Swin Transformer(C3STR) [22].
- *Етап 3: Посилення злиття цільових характеристик моделі.* Це буде досягнуто шляхом додавання механізму уваги GAM [26] у модуль Concat.

У наступних підрозділах детально розглянуті запропоновані кроки методу.

#### 2.1 Особливості моделі глибокого навчання YOLOv8

YOLOv8 є еволюцією попередніх версій YOLO, яка включає нові характеристики та вдосконалення для підвищення продуктивності мережі. Ця версія є останньою у серії YOLO та досягає високої швидкості та ефективності у розпізнаванні об'єктів. YOLOv8 пропонує п'ять моделей різного розміру: nano, small, middle, large та extra-large. Модель nano має лише 3,2 мільйони параметрів, що сприяє легкому впровадженню на мобільні пристрої та пристрої, які працюють лише на CPU. Для досягнення оптимального балансу між точністю детекції та швидкодією

системи, було вирішено використовувати модель YOLOv8s для задачі виявлення багатороторних БПЛА. Ця модель є результатом розширення та поглиблення структури nano моделі. У Таблиці 2.1 наведено порівняння моделей YOLO.

Таблиця 2.1 – Порівняння моделей YOLOv8.

Модель	Розмір зображення	mAP	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Кількість параметрів (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Структуру моделі YOLOv8 можна поділити на backbone, neck та head. Backbone мережі YOLO служить для витягування ознак з вхідного зображення. Він зазвичай складається з послідовності конволюційних шарів. Head мережі YOLO відповідає за кінцеву класифікацію та локалізацію об'єктів. В цьому компоненті використовуються отримані ознаки для передбачення класів об'єктів та їх розташувань на зображенні. Head складається з конволюційних шарів, які видають кінцеві вихідні дані, такі як обмежувальні прямокутники, класи об'єктів та їх вірогідності. Neck є додатковим шаром між backbone та head, який допомагає в уточненні та поліпшенні ознак, отриманих з backbone. Цей компонент часто використовує модулі агрегації ознак для збалансування та комбінації інформації з різних шарів backbone. Завдання backbone - зменшити просторові розміри зображення, при цьому зберігаючи важливу інформацію про ознаки. Дизайн мережі YOLOv8 представлений на Рисунку 2.1.

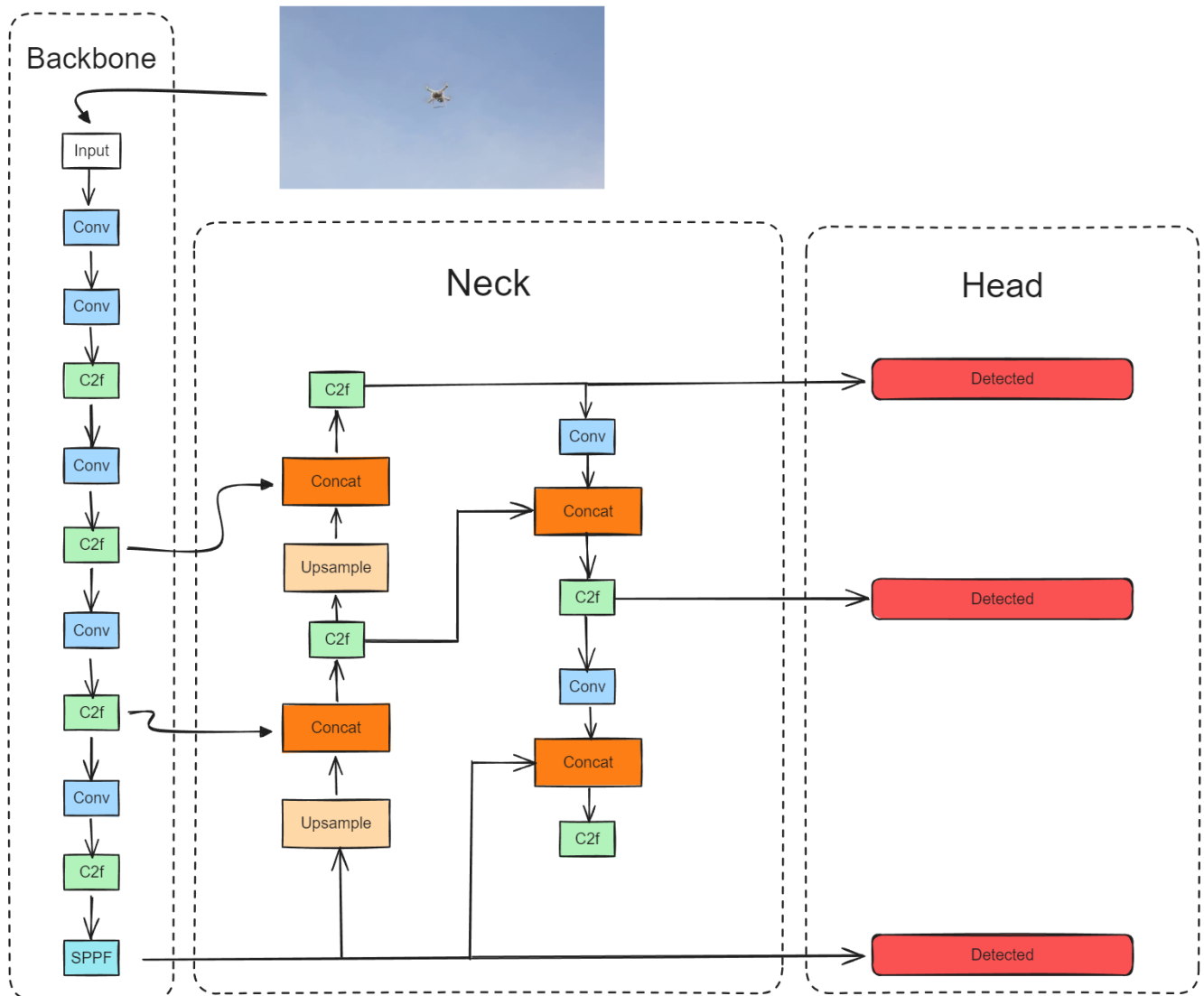


Рисунок 2.1 – Структура мережі YOLOv8

YOLOv8 працює без використання якорів (Рисунок 2.2). Це означає, що прогнозується центр об'єкта, а не зміщення від відомого якорного блоку. У попередніх версіях YOLO якорні блоки були відомою складною частиною, оскільки вони могли представляти розподіл блоків цільового бенчмарку, але не розподіл призначеного для користувача набору даних.

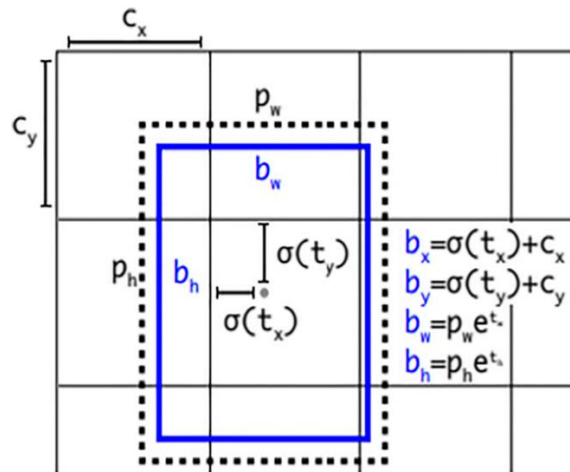


Рисунок 2.2 – Візуалізація якірного блоку в YOLO [27]

Безякірне виявлення зменшує кількість прогнозів, що пришвидшує процес немаксимального пригнічення (NMS), складний етап постобробки, який відбирає кандидатські детекції після виведення. В кінцевому підсумку, вона виконує багатомасштабні прогнози, використовуючи ознаки, зменшені у  $8\times$ ,  $16\times$ , та  $32\times$ , для досягнення точних прогнозів для малих, середніх і великих цілей. Детальні модулі в мережі YOLOv8 ілюстровані на Рисунку 2.3.

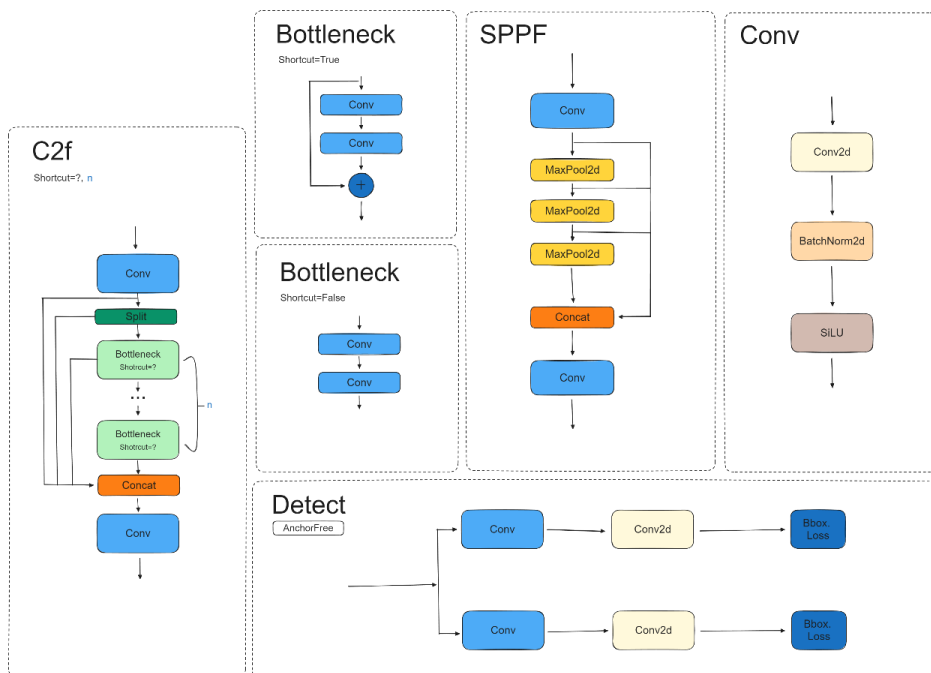


Рисунок 2.3 – Структурна схема мережі YOLOv8

Вивчення структури моделі YOLOv8 дозволило виявити ключові аспекти, які можуть підвищити ефективність та точність виявлення багатороторних БПЛА. Основні напрямки вдосконалення включають покращення здатності моделі до виявлення малих об'єктів, інтеграцію нових технологій, таких як Swin Transformer у backbone, та впровадження механізму уваги для покращення процесу злиття цільових характеристик. Ці інновації можуть значно збільшити потенціал моделі YOLOv8 у задачі виявлення малих БПЛА, забезпечуючи високу точність при зменшеному обчислювальному навантаженні. Використання безякірного виявлення дозволяє спростити модель та прискорити процес виявлення, а також робить її більш адаптивною до різних розмірів цілей. У подальших дослідженнях буде розглянуто можливість інтеграції трансформерних підходів у структуру YOLOv8 для подальшого підвищення ефективності та точності виявлення БПЛА.

## 2.2 Покращення здатності моделі до виявлення малих цілей

При використанні камери для захоплення зображення БПЛА, з метою уникнення виходу літаючих апаратів з поля зору, зазвичай обирається камера, що має широкий кут огляду. Це призводить до того, що пропорція БПЛА на зображенні є відносно малою. Оригінальна мережева структура YOLOv8 зменшує зображення в цілому у п'ять разів, щоб отримати п'ять шарів вираження характеристик ( $P1$ ,  $P2$ ,  $P3$ ,  $P4$  та  $P5$ ), де  $P_i$  позначає розмір зображення зменшений у  $\frac{1}{2^i}$  від оригінального зображення. Хоча багатомасштабне злиття характеристик досягається в частині мережі неск, через верхні та нижні шляхи агрегації, це не впливає на масштаб карти характеристик, і фінальна частина head виявляється після проходження через шари  $P3$ ,  $P4$  і  $P5$ . Масштаби карти характеристик відповідно становлять  $80 \times 80$ ,  $40 \times 40$  та  $20 \times 20$ . У завданні виявлення маленьких цілей часто бувають крихітні цілі для виявлення. Зображення, отримані з камери, часто включають в себе множину дрібних цілей БПЛА, що зазвичай мають розміри менше ніж  $10 \times 10$  пікселів. Такі об'єкти втратили більшу частину своєї інформації про характеристики після кількох

зменшень і все ще важко виявляються з високою роздільною здатністю шаром детекції P3.

З метою покращення ідентифікації малих цілей та підвищення ефективності виявлення БПЛА, було запропоновано доповнити архітектуру моделі YOLOv8 новим шаром у компоненті head, використовуючи характеристики шару P2, як це відображено на Рисунку 2.4. Роздільна здатність шару P2 становить  $160 \times 160$  пікселів, що еквівалентно лише двом операціям зменшення в основній мережі, що містить більшу інформацію про базові характеристики цілі. Дві характеристики шару P2, отримані зверху вниз і знизу вгору в мережі шиї, злиті з характеристиками того ж масштабу в основній мережі, у формі Concat, тоді як вихідні характеристики є злитими результатами трьох вхідних характеристик, що робить голову детекції шару P2 швидкою та ефективною при роботі з крихітними цілями. Голова детекції шару P2, разом з оригінальною head, може ефективно пом'якшувати зміни масштабу, спричинені шаром детекції P2, який, разом з початковою head, може ефективно зменшити негативні ефекти змін масштабу. Доданий шар head є специфічним для базових характеристик і генерується з карт характеристик низького рівня, які більш чутливі до маленьких цілей. Хоча додавання цього детекційного шару head призводить до збільшення обчислювального навантаження моделі і як наслідок збільшує загальний об'єм використання пам'яті. Але це значно покращує виявлення крихітних цілей.

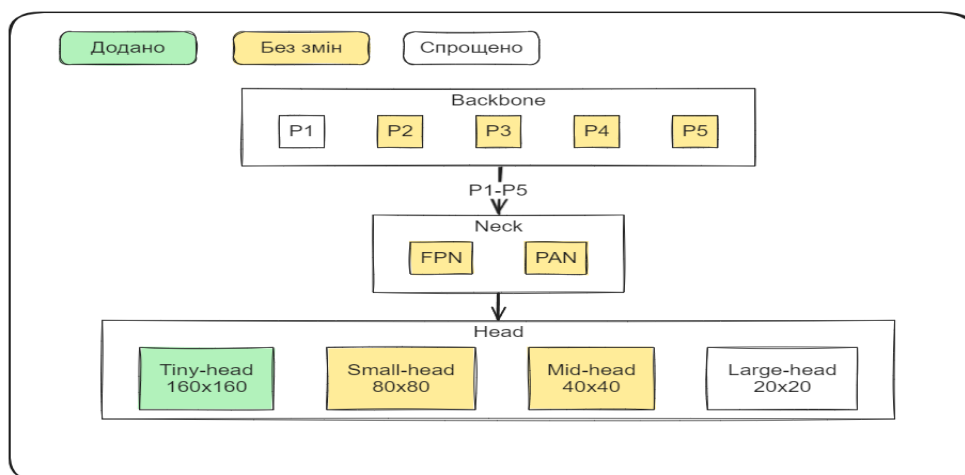


Рисунок 2.4 – Вдосконалена структура head

### 2.3 Видалення детекційного шару для виявлення великих цілей

Детекційний шар виявлення великих цілей, шар *P5*, отримується шляхом зменшення зображення у 32 рази. Коли розмір цілі менший за 32 пікселі, існує висока ймовірність, що при вибірці буде захоплено лише одну точку цілі або вона не буде захоплена зовсім. Таким чином, шар *P5* є зайвим при детекції малих БПЛА. На підставі проведеного аналізу, стало очевидним, що для підвищення ефективності мережі YOLOv8 необхідно виключити з її структури шар, відповідальний за прогнозування великих цілей, а також пов'язані з ним шари, що відповідають за екстракцію та злиття характеристик.

Зберігаються лише карти характеристик з зменшенням у 4, 8 і 16 разів для прогнозування БПЛА. У вдосконаленій структурі мережі, показаній на Рисунку 2.3, карти ознак, зменшені у 16 разів на третьому шарі C2f, безпосередньо подаються в SPPF для отримання багатомасштабних ознак. Потім злиті карти ознак викидаються з модулю Upsample-Concat-C2f і безпосередньо підключаються до наступного модуля, а всі шари мережі після середньої голови детекції відкидаються. Така вдосконалена структура мережі зменшує обчислювальне навантаження, усуваючи зайві розрахунки, і забезпечує гарантовану точність.

Детальний аналіз детекційних шарів мережі YOLOv8 демонструє необхідність у їх модифікації. Виявлено, що детекційний шар *P5*, призначений для великих цілей, не є ефективним для малих БПЛА через ризик недостатньо точної вибірки ознак або втрати цілі. Таким чином, оптимізація мережі полягає в вирізанні зайвих шарів, зокрема шару прогнозування великих цілей та пов'язаних з ним шарів екстракції та злиття характеристик. У результаті цього вдосконалення, мережа фокусується на картах характеристик, зменшених у 4, 8 і 16 разів, що забезпечує більш точне виявлення малих БПЛА із зниженням обчислювального навантаження. Така модифікована структура не тільки покращує точність виявлення малих цілей, але й забезпечує більшу ефективність обчислень, що є критично важливим для систем, що вимагають швидкої реакції та високої продуктивності.

## 2.4 Трансформери

Трансформерна архітектура є видом штучної нейронної мережі, розробленої для аналізу контексту і виявлення зв'язків у послідовних даних. Ця технологія здобула популярність завдяки своєму ефективному використанню в сферах мови і перекладу [34], де вона вперше демонструвала свої вражаючі можливості. Завдяки введенню трансформерів у галузі комп'ютерного зору в 2020 році [33], вони стали ще більш значимими. Недавні дослідження виявили, що трансформери, використані в завданнях комп'ютерного зору, показують результати на рівні з CNN, інколи навіть перевершуючи їх [35], за умов наявності достатньої кількості тренувальних даних.

Трансформерна структура має дві основні частини: кодер та декодер. Кодер обробляє вхідну послідовність, створюючи проміжне представлення, яке потім передається декодеру для генерування вихідної послідовності. Вони складаються з позиційного вбудовування, за яким слідує шар самоуваги та повністю з'єднані шари з активаційною функцією ReLU. Декодер об'єднує вихід останнього блоку кодера з виходом попереднього замаскованого шару з множинною увагою, що дозволяє ефективно обробляти інформацію.

Як було зазначено раніше, основою роботи трансформера є обробка послідовностей. У класичній моделі трансформера ці послідовності складаються з речень, поділених на слова. Ці слова трансформуються у токени, які далі вбудовуються у вектори з виміром  $d$  за допомогою спеціально навчених вбудовувань. Додавання позиційної інформації до цих векторів дозволяє моделі розпізнавати і враховувати порядок слів у реченні.

В оригінальній конструкції трансформера позиційне кодування виконується на основі функцій синуса та косинуса, які описані в (2.1, 2.2). У цих рівняннях  $p$  означає позицію слова в реченні, а  $i$  - відповідну вимірність. Розмірність цього позиційного вбудовування відповідає розмірності  $d$  звичайного вбудовування, що дозволяє їм ефективно комбінуватися.

$$Pos(p, 2i) = \sin\left(\frac{p}{10000^{2i/d}}\right), \quad (2.1)$$

$$Pos(p, 2i + 1) = \cos\left(\frac{p}{10000^{2i/d}}\right), \quad (2.2)$$

Центральним елементом архітектури трансформера є механізм, відомий як механізм самоуваги. Цей метод дозволяє моделі визначати ступінь зв'язку або схожості між компонентами вхідної послідовності. Основний принцип трансформера полягає у використанні так званої точкової уваги до продукту, що реалізується через процес відображення запиту ( $Q$ ) і пари ключ-значення ( $K, V$ ) у вихідний результат. Цей процес здійснюється за допомогою зваженого підсумовування, де ваги визначаються на основі ступеня кореляції між запитом і відповідним ключем.

Для формування значень  $Q, K, V$  використовується метод множення вхідного вкладу  $X$  на три окремі вагові матриці:  $WQ, WK, WV$ . Розрахунок значення уваги виконується згідно з рівнянням 3, до якого додається масштабний коефіцієнт -  $\sqrt{d_k}$ . Цей коефіцієнт важливий для запобігання надмірного зростання величин векторів у процесі обчислень.

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.3)$$

Функція  $\text{softmax}$ , яка визначена у формулі (2.4), використовує стандартну експоненційну функцію  $e^x$ . Цю функцію можна розуміти як процес перетворення вхідних даних у розподіл ймовірностей. Функція  $\text{softmax}$  масштабує кожний елемент вхідного вектора, перетворюючи його на ймовірнісне значення, що відображає його відносну значимість в контексті всього набору даних.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}, \quad (2.4)$$

Розрахована за допомогою  $\text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right)$  матриця може бути розглянута як індикатор частини вбудовування вхідних даних, на які потрібно зосередити увагу. Тоді як матриця  $V$  представляє значення, які мають бути виведені у результаті. Лінійне поєднання цих двох матриць формує кінцевий вихідний результат.

Концепція багатоголової уваги (multi-head attention) є розширенням цієї ідеї. Замість використання одного обчислення уваги з  $Q, K, V$  одного виміру  $d$ , можна виконати  $h$  різних обчислень уваги, кожне з власними  $Q_i, K_i, V_i$  розмірностей  $d_q, d_k, d_v$ . Кожен окремий результат цих обчислень,  $Z_i$ , називається "головою". Ці голови є незалежними одна від одної і можуть обчислюватися паралельно. Потім ці голови конкатенуються і проходять через вагову матрицю  $W^O$  розміром  $d \times d$ , де  $d = h * d_k$ . Детальний опис цього процесу можна знайти у відповідних рівняннях.

$$Q_i = XW_i^Q, \quad (2.5)$$

$$K_i = XW_i^K, \quad (2.6)$$

$$V_i = XW_i^V, \quad (2.7)$$

$$Z_i = \text{Attention}(Q_i, K_i, V_i), \quad i = 1 \dots h, \quad (2.8)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(Z_1, Z_2, \dots, Z_h)W^O, \quad (2.9)$$

Причина введення багатоголової уваги в архітектуру трансформера полягає в тому, що вона надає моделі здатність одночасно зосереджуватись на різних аспектах

інформації з різних представлень та позицій [34]. Це відіграє ключову роль, оскільки з одною головою уваги процес усереднення може перешкоджати цій багатоаспектності.

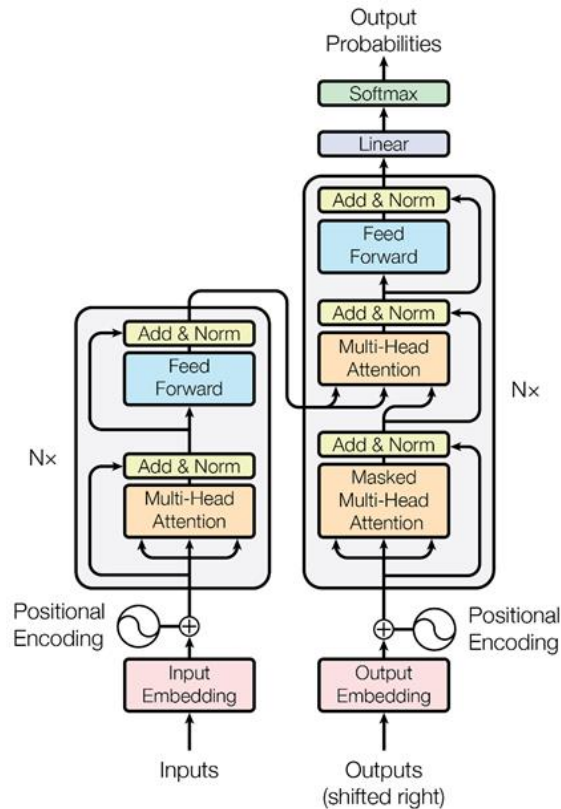
У декодері, механізм уваги доповнений маскуванням. Вкладення, що були створені до цього моменту, маскуються за допомогою матриці  $M$ , яка складається з нулів і дуже малих значень для елементів, що повинні бути приховані. Це дозволяє уникнути кореляції між замаскованими запитами та відповідними значеннями, що детально описано в (2.10).

$$\text{MaskedAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V, \quad (2.10)$$

Кожен шар трансформера супроводжується етапом "Додати і нормалізувати" (Add & Norm), зображеним на Рисунку 2.5. Ця процедура включає додавання залишкових зв'язків до попереднього підшару і нормалізацію цього шару, яка працює подібно до пакетної нормалізації, але обчислення середнього значення та дисперсії виконується окремо для кожного каналу та вектора.

Після послідовності кодер-декодер зазвичай розміщується серія повністю з'єднаних шарів, які функціонують як класифікаційна або регресійна голова моделі.

Аналіз трансформерної архітектури виявляє, що спочатку вона була розроблена для обробки мовних послідовностей, з конвертацією слів та речень у токени та їх вбудовуванням у вектори. Однак, цей підхід може бути адаптований для виявлення зв'язків між об'єктами у зображеннях, що робить його перспективним у сфері комп'ютерного зору. Ключовим моментом тут є перенесення цих технік обробки послідовностей на аналіз візуальних даних, де елементи зображення можуть бути подані у вигляді токенів, аналогічно до слів у реченні. В подальших розділах буде розглянуто застосування цих технік для виявлення та класифікації об'єктів у зображеннях.



Рисунку 2.5 - Оригінальна структура трансформеру [34]

Поєднання цих методів із традиційними підходами, такими як CNN, може привести до розвитку гібридних мереж, що забезпечать ефективніше виявлення та класифікацію об'єктів, включаючи БПЛА.

### 2.4.1 Архітектура трансформера зору

Запровадження трансформера зору (vision transformer) спричинило зростання зацікавленості в використанні трансформерів для рішення завдань у галузі комп'ютерного зору. Хоча архітектура трансформера зору має багато спільного з оригінальним трансформером, існують певні відмінності. Ключовою особливістю трансформера зору є використання лише кодерної частини трансформера, відмовляючись від декодера. Це означає, що трансформер зору займається створенням карти об'єктів, але не її декодуванням, реалізуючи це через інтеграцію спеціалізованої голови на кінці кодера.

Ця архітектура виявляється ефективною і гнучкою, дозволяючи адаптувати трансформер зору для різних типів завдань. Замість перетворення речень у послідовність слів, трансформер зору ділить зображення на послідовність фрагментів. Ці фрагменти перетворюються у площину і вбудовуються у d-вимірний простір за допомогою навченої лінійної проєкції. Для вводу цих фрагментів у кодер, до них застосовуються позиційні коди.

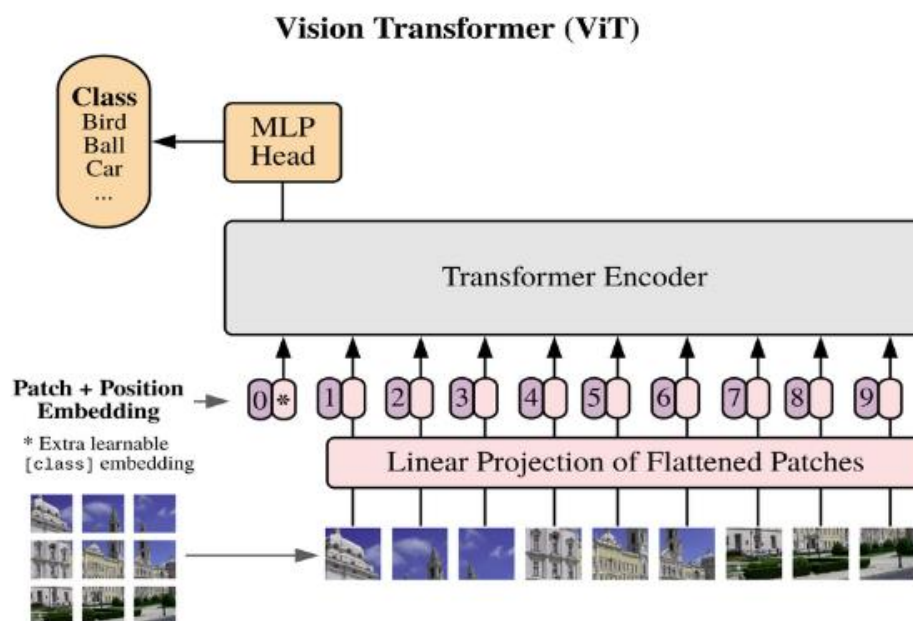


Рисунок 2.6 - Архітектура трансформера зору [32]

У трансформері зору всі шари повної зв'язності, відомі як багат шарові перцептрони (MLP), використовують унікальну функцію активації під назвою лінійна одиниця гауссової похибки, або GELU (Gaussian error linear unit). Ця функція активації, розроблена Хендриксом і Гімпелом [36], стала широко вживаною в різноманітних трансформерних архітектурах. GELU відрізняється від традиційної ReLU тим, що вона регулює входи на основі їхніх абсолютних значень, а не просто від'ємності чи додатності.

Трансформер зору, який широко застосовується в галузі класифікації зображень, стикається з певними викликами, зокрема через потребу в обробці високої роздільної здатності зображень, що суттєво відрізняється від завдань, пов'язаних з обробкою тексту. Основна проблема при використанні трансформерів в аналізі зображень полягає в тому, що традиційні трансформери зору мають квадратичну складність обчислень щодо розміру зображення через механізми глобальної самоуваги, що вимагає значних обчислювальних ресурсів.

Одним з методів оптимізації та забезпечення більш ефективного вилучення ознак з зображень є використання Swin Transformer [37]. Цей підхід внесений у розробку з метою подолання обмежень стандартних трансформерів, створюючи можливість більш ефективної обробки зображень з високою роздільною здатністю.

#### **2.4.2 Архітектура Swin Transformer**

Swin Transformer [33], відомий своєю передовою продуктивністю в області комп'ютерного зору, відрізняється від своїх попередників, зокрема від трансформера зору (ViT) [32], високою ефективністю та точністю. Ці якості зробили Swin Transformer популярним вибором для використання в сучасних моделях комп'ютерного зору.

Swin Transformer застосовується в широкому спектрі задач комп'ютерного зору, зокрема у таких складних завданнях, як виявлення об'єктів у наборі даних COCO. Він використовує модифіковану версію механізму самоуваги трансформера (MSA), відому як віконна багатоголова самоувага (W-MSA). Цей підхід полягає в обчисленні уваги в межах локальних вікон, з подальшим переміщенням цих вікон між шарами самоуваги. Swin Transformer також об'єднує фрагменти зображення на більш глибоких шарах, дозволяючи обчислювати увагу ієрархічно та уникаючи лінійної складності відносно розміру вхідних даних. В якості функції активації використовується GELU.

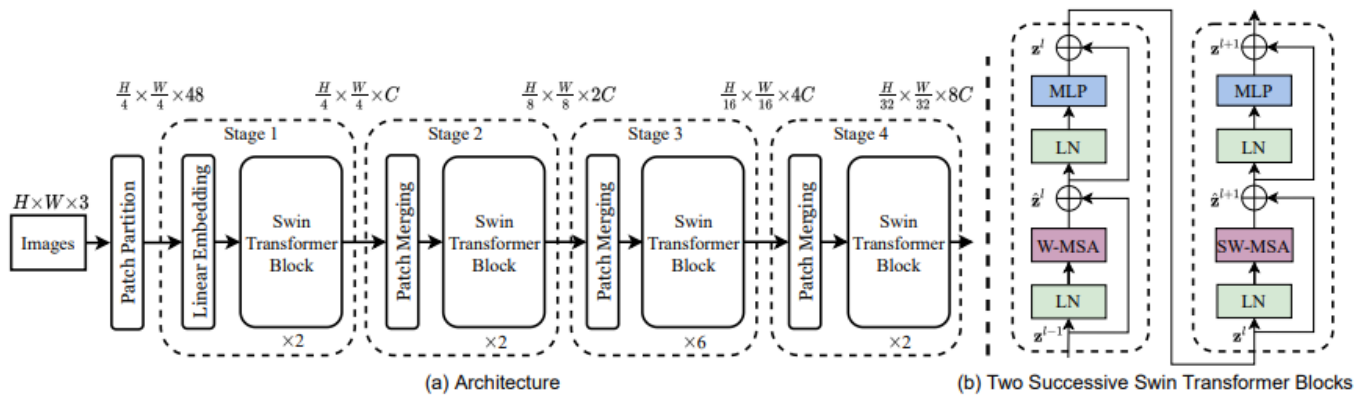


Рисунок 2.7 – Загальна архітектура Swin трансформеру [33]: а – Архітектура Swin Transformer (Swin-T); б – два послідовних блоки Swin Transformer

Одними з ключових нововведень Swin Transformer є ієрархічні карти функцій та зміщене вікно уваги, що допомогли вирішити проблеми, з якими зіткнувся оригінальний трансформер зору. Назва "Swin" походить від «Shifted Window Transformer», що відображає його унікальний підхід до обчислення уваги. Загальна архітектура Swin Transformer детально представлена на Рисунку 2.7.

Такий підхід дозволяє забезпечити більшу гнучкість та ефективність при обробці зображень, особливо для завдань, які вимагають великого обсягу даних з високою роздільною здатністю.

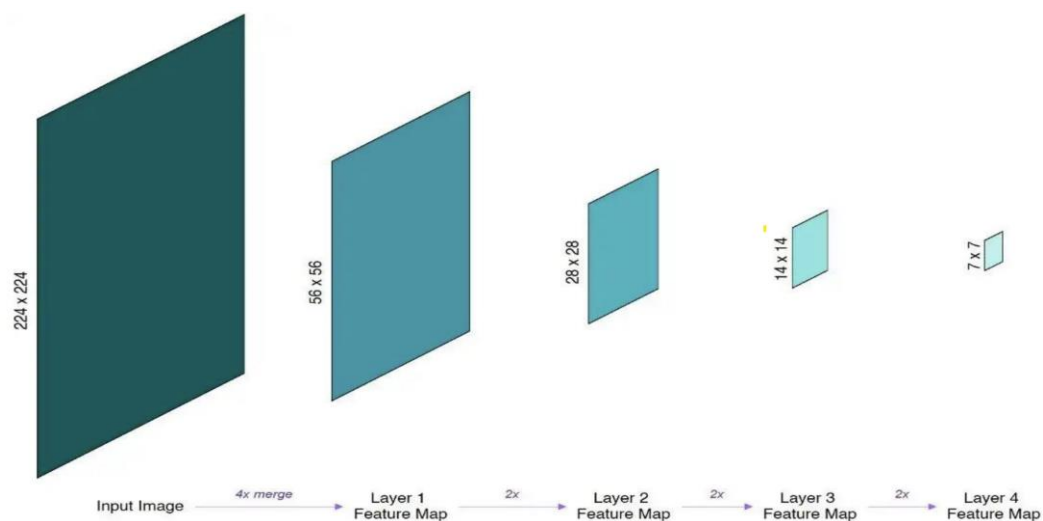


Рисунок 2.8 – Карти ієрархічних функцій у Swin Transformer (глибина карти функцій опущена з метою спрощення) [32]

На Рисунку 2.8 зображено як Swin Transformer використовує унікальний підхід, де карти функцій поступово об'єднуються та зменшуються після кожного шару, що призводить до формування ієрархічної структури цих карт. Просторова роздільна здатність цих ієрархічних карт функцій узгоджується з роздільною здатністю в мережах ResNet, що було зроблено навмисне для забезпечення зручності заміни Swin Transformer замість мереж ResNet у різноманітних методах комп'ютерного зору. Використання ієрархічних карт функцій дозволяє Swin Трансформерам бути ефективними в завданнях, де важливе детальне передбачення, таких як семантична сегментація. На відміну від цього, трансформер зору використовує карти функцій з єдиною, низькою роздільною здатністю у своїй архітектурі [38].

Swin Transformer впроваджує техніку зменшення дискретизації без згортки, відому як злиття патчів. У своєму трансформерному блоку, Swin Transformer замінює стандартний багатоголовий модуль самоуваги (MSA), який використовується у трансформері зору, на модуль MSA з вікном (window-based multi-head self-attention / W-MSA) і модуль MSA зі зміщеним вікном (shifted window multi-head self-attention / SW-MSA).

На Рисунку 2.9 зображено модулі блоку Swin Transformer.

Блок Swin Transformer складається з двох основних під-блоків. Кожен з цих під-блоків починається з рівня нормалізації, за яким слідує модуль уваги, а потім ще один рівень нормалізації та рівень MLP. Перший під-блок застосовує модуль Window MSA (WMSA), а другий - Shifted Window MSA (SW-MSA).

Стандартний MSA, який використовується у трансформері зору, проводить глобальний самоконтроль, обчислюючи зв'язки між кожним патчем і всіма іншими патчами, що призводить до квадратичної складності відносно кількості патчів. Це робить його менш підходящим для зображень високої роздільної здатності.

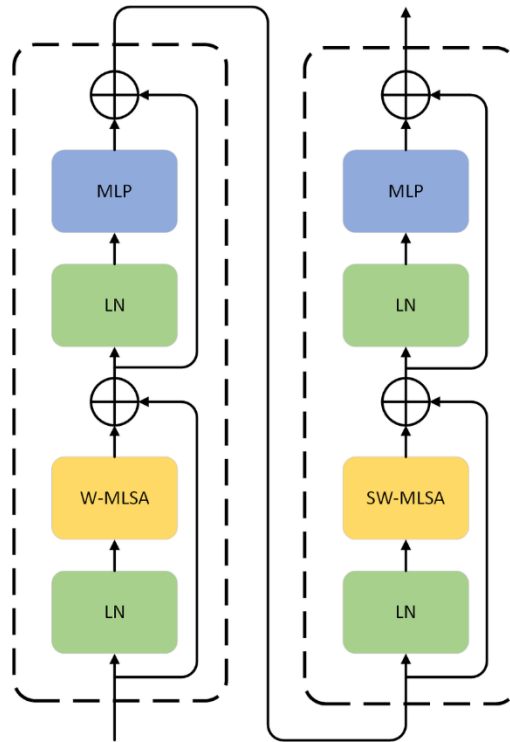


Рисунок 2.9 – Блок Swin Трансформер, з 2 під-блоками (W-MSA та SW-MSA) [32]

Swin Transformer використовує підхід на основі віконного MSA, де увага обчислюється всередині кожного вікна окремо. Так як розмір вікна є фіксованим у всій мережі, складність обчислень у цьому випадку є лінійною відносно кількості патчів. Це значно покращує ефективність у порівнянні зі стандартним MSA.

Однак, віконний MSA обмежує самоувагу до окремих вікон, що може обмежувати загальну ефективність моделі. Swin Трансформер вирішує цю проблему, використовуючи модуль SW-MSA після W-MSA, який зміщує вікна на певну величину для створення міжвіконних зв'язків. Це зміщення може призвести до "загублених" патчів, але Swin Трансформер вирішує цю проблему за допомогою "циклічного зсуву", переносячи ці патчі до вікон з неповними патчами. Цей підхід дозволяє ввести важливі перехресні зв'язки між вікнами та покращує загальну продуктивність мережі.

Ієрархічні карти функцій та зміщене вікно MSA дозволили Swin Трансформеру ефективно вирішити недоліки оригінального трансформера зору. В

результаті, Swin Трансформер широко використовується в різноманітних завданнях комп'ютерного зору, включаючи класифікацію зображень та виявлення об'єктів.

## 2.5 Вдосконалення модуля злиття ознак

Global attention mechanism (GAM) – це легкий, практичний та простий компонент, який можна безпроблемно інтегрувати в архітектури CNN. Його основна мета полягає в підсиленні продуктивності глибоких нейронних мереж за рахунок мінімізації втрати інформації та посилення глобальної взаємодії представлень в рамках отриманої карти ознак. Модуль GAM використовує механізм уваги CBAM, який працює від каналу до просторового порядку. У роботі [39] модуль GAM був успішно інтегрований в різні моделі на різних наборах даних та завданнях класифікації, що призвело до значного поліпшення продуктивності моделі, що підкреслює ефективність модуля GAM. Як модуль типу "plug-and-play", GAM широко цитується, як у літературі [40], шляхом вставки GAM в backbone та head YOLOv7, що дозволяє мережі виділяти критичні ознаки шляхом посилення взаємодії глобальних вимірних ознак. Структура GAM показана на Рисунку 2.10.

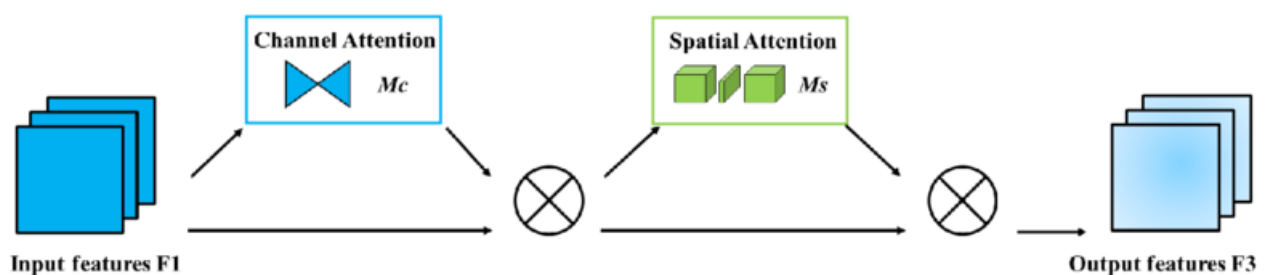


Рисунок 2.10 – Модуль уваги GAM [41]

З огляду на відображення вхідного атрибуту  $F_1$ , проміжні стани  $F_2$  і вихід  $F_3$  визначаються наступним чином:

$$F_2 = M_c(F_1) * F_1, \quad (2.11)$$

$$F_3 = M_s(F_2) * F_2, \quad (2.12)$$

Оскільки малі цілі мають невеликі розміри, а також обмежені та неяскраві ознаки, введення модуля уваги GAM до частини злиття ознак може посилити глобальну взаємодію та покращити здатність мережі до збереження ознак малих цілей, одночасно вдосконалюючи злиття ознак у частині неск мережі. У завданні детекції модуль уваги GAM може допомогти моделі ефективно виділяти регіон уваги та покращувати продуктивність детекції.

## 2.6 Статистичні показники для оцінювання якості отриманої моделі глибокого навчання

Щоб перевірити продуктивність моделі, в якості експериментальних показників оцінювання вибрано точність ( $P$ ), повноту ( $R$ ), середню точність ( $AP$ ), середню точність середніх значень ( $mAP$ ), кількість параметрів, розмір моделі та кадри в секунду (FPS).

Чутливість алгоритму класифікації (точність –  $P$ ) та специфічність алгоритму класифікації (повнота –  $R$ ) розраховуються таким чином:

$$P = \frac{TP}{(TP + FP)} \cdot 100\%, \quad (2.13)$$

$$R = \frac{TP}{(TP + FN)} \cdot 100\%, \quad (2.14)$$

де основними показниками є True Positive ( $TP$ ), False Positive ( $FP$ ), і False Negative ( $FN$ ).  $TP$  - це істинно додатні випадки, де класифікатор правильно ідентифікує позитивний випадок.  $FP$  - це хибно додатні випадки, коли класифікатор помилково ідентифікує негативний випадок як позитивний (помилка I роду).  $FN$  - це помилково

від'ємні випадки, коли класифікатор помилково ідентифікує позитивний випадок як негативний (помилка II роду).

Точність (P) вимірює, яка частка об'єктів, визначених класифікатором як позитивні, дійсно є позитивними. Повнота (R) показує, яку частку позитивних випадків з усіх існуючих позитивних класифікатор зміг виявити.

На основі отриманих даних можна побудувати криву помилок (ROC крива) - це графік, який відображає співвідношення між точністю та повнотою класифікаційного алгоритму. AUC (Area Under the Curve) - площа під ROC кривою, яка є кількісною мірою якості класифікатора. Чим більша площа AUC, тим краще якість класифікатора. Значення AUC, що дорівнює 0,5, вказує на відсутність дискримінаційної здатності класифікатора (аналогічно випадковому вгадуванню), в той час як значення менше 0,5 свідчить про те, що класифікатор працює гірше, ніж випадкове вгадування.

Середня точність (AP) та середня точність середніх значень (mAP) розраховуються таким чином:

$$AP = \int_0^1 p(r) d(r), \quad (2.15)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (2.16)$$

де  $N$  — кількість категорій, а  $AP$  — середня точність кожної категорії. У нашому завданні виявлення БПЛА,  $N = 1$ .

## Висновки до розділу 2

У розділі запропоновано покращену схему розпізнавання об'єктів малих розмірів. За основу було обрано нейромережу YOLOv8. Було розглянуто стандартну структуру моделі та запропоновано нову з інтеграцією Swin Transformer. Цей підхід поєднує переваги YOLOv8, відомої своєю швидкістю і точністю в задачах об'єктного розпізнавання, з високою здатністю трансформерів аналізувати контекст та виявляти складні зв'язки в даних. Така інтеграція спрямована на підвищення точності розпізнавання БПЛА, особливо в умовах різноманітності фонів та змін умов освітлення. Swin Transformer, інтегрований у цю систему, дозволяє глибше аналізувати просторові зв'язки між об'єктами в кадрі, що є критичним для точного виявлення БПЛА в складних сценаріях. Ця технологія відкриває двері до більш ефективного моніторингу повітряного простору та посилення безпекових заходів.

Також було покращено здатність моделі до виявлення крихітних цілей. Для цього було додано детекційний шар P2 до модуля head. Шар що відповідає за виявлення великих об'єктів було видалено.

Розглянуто global attention mechanism, який покращив продуктивність моделі.

## **РОЗДІЛ 3**

### **Проектування структури інформаційної системи для експериментального дослідження запропонованого методу**

Для тестування запропонованого у попередньому розділі методу спроектована інформаційна система, основою якої є нейронна мережа YOLOv8.

Запропонована система є інтеграцію передових методів глибокого навчання та комп'ютерного зору. Використання моделі YOLOv8, відомої своєю високою продуктивністю та точністю, стало ключовим вибором у проектуванні цієї системи. Основною метою є не лише обробка та аналіз зображень, але й точне виявлення та класифікація БПЛА, що має велике значення в сучасних реаліях.

Розробка системи охоплює різні аспекти: від структурних компонентів і архітектури до методів збору даних, передобробки, тренування та оптимізації моделі. Важливою частиною роботи стане аналіз результатів експериментальних досліджень, що дозволить оцінити ефективність запропонованого підходу.

Проект демонструє, як можна застосувати новітні досягнення у галузі глибокого навчання для розв'язання конкретних практичних задач. Результати цієї роботи відкривають нові можливості для розвитку та вдосконалення методів комп'ютерного зору, що можуть бути використані у різних сферах, від безпеки до моніторингу довкілля.

Далі опишемо основні елементи та засоби, які використовувались для проектування ІС.

#### **3.1 Засіб розробки та тестування моделей машинного та глибокого навчання Python**

Для розробки ІС пропонується використати інтерпретовану об'єктно-орієнтовану мову програмування високого рівня із суворою динамічною типізацією. Вибір наведеного засобу впливає із поширеності його для розв'язку подібних задач.

Ключовими характеристиками Python є:

- підтримка об'єктно-орієнтованого програмування: Python дозволяє створювати розширені програмні структури, що сприяє складному програмуванню;
- інтерпретована природа: Python компілюється на льоту, що забезпечує гнучкість і швидку розробку;
- розгорнута стандартна бібліотека: Python включає широкий спектр корисних модулів, спрощуючи розробку;
- динамічна типізація: автоматичне визначення типів даних в Python зменшує помилки та прискорює написання коду;
- переносимість: Python можна використовувати на різних системах, що робить його ідеальним для розробки крос-платформних програм.

Python є популярним варіантом у галузях машинного навчання, веб-розробки та науки про дані, завдяки своїй універсальності та легкості використання.

### 3.2 Пакет Ultralytics для використання YOLOv8

Пакет Ultralytics для Python відомий як YOLOv8, це сучасна модель, що вдосконалює попередні версії YOLO, вносячи нові функції та поліпшення. YOLOv8 розроблений для швидкості, точності та простоти використання, що робить його придатним для різноманітних завдань, включаючи виявлення та відстеження об'єктів, сегментацію, класифікацію зображень і визначення пози.

Пакет можна легко встановити в середовищі Python  $\geq 3.8$  з PyTorch  $\geq 1.8$ , використовуючи команду *pip install*.

Щодо використання, YOLOv8 підтримує інтерфейс командного рядка (CLI) та середовище Python. У CLI його можна використовувати з простою командою *yolo* для різних завдань, приймаючи додаткові аргументи за потреби. Наприклад, для передбачення за допомогою моделі на джерелі зображень команда буде *yolo predict model=yolov8n.pt source='https://ultralytics.com/images/bus.jpg'*. У середовищі Python пакет YOLOv8 можна імпортувати, а моделі - завантажувати, навчати, оцінювати та

експортувати. Це включає створення нових моделей з нуля, завантаження готових моделей і експорт моделей у різні формати, наприклад, ONNX.

Моделі YOLOv8 постачаються з попередньо навченими наборами даних, такими як COCO для завдань виявлення, сегментації та визначення поз, а також ImageNet для завдань класифікації. Усі моделі виявлення, сегментації та поз також підтримують режим відстеження.

Ultralytics інтегрується з провідними платформами ШІ, покращуючи маркування даних, навчання, візуалізацію та управління моделями. Ці інтеграції включають Roboflow для маркування та експорту користувацьких наборів даних, ClearML для відстеження, візуалізації та віддаленого навчання моделей, Comet для збереження моделей та відновлення навчання, а також Neural Magic для прискорення виводу YOLOv8 до 6 разів.

YOLOv8 – це не просто чергова модель виявлення об'єктів, але й універсальний фреймворк, який охоплює весь життєвий цикл машинного навчання, від збору даних і тренування моделі до її валідації, розгортання та використання у реальному світі. Кожен режим роботи призначений для конкретних завдань і надає потрібну гнучкість та ефективність для різних сценаріїв використання.

Можна виділити наступні режими роботи YOLOv8:

- Train mode: Призначений для тонкого налаштування моделі на власних або завантажених наборах даних.
- Val mode: Використовується після тренування для перевірки продуктивності моделі.
- Predict mode: Розкриває прогностичні можливості моделі на реальних даних.
- Export mode: Готує модель до розгортання у різних форматах.
- Track mode: Перетворює модель виявлення об'єктів на інструмент для реального слідкування.

– **Benchmark mode:** Аналізує швидкість та точність моделі у різних умовах розгортання.

У режимі **Train** модель YOLOv8 навчається на конкретному наборі даних. Процес тренування включає оптимізацію параметрів моделі для точного виявлення класів та місцезнаходження об'єктів на зображеннях.

Режим **Val** використовується для оцінювання моделі YOLOv8 після навчання. Модель перевіряється на валідаційному наборі даних для вимірювання її точності та спроможності узагальнювати. Цей режим може бути використаний для налаштування гіперпараметрів моделі.

У режимі **Predict** використовується навчена модель YOLOv8 для аналізу нових зображень або відео. Модель завантажується з файлу контрольної точки, і користувач може подавати зображення або відео для інференції. Модель визначає класи та місцезнаходження об'єктів у вхідних даних.

Режим **Export** використовується для перетворення моделі YOLOv8 у формат, який можна використовувати для розгортання. Модель конвертується у формат, сумісний з іншими програмними додатками або апаратними пристроями, що корисно для впровадження моделі у виробничі середовища.

У режимі **Track** використовується модель YOLOv8 для відстеження об'єктів у реальному часі. Модель завантажується з файлу контрольної точки, і користувач може подавати відеопотік для відстеження об'єктів у реальному часі. Цей режим корисний для систем спостереження або автономних автомобілів.

У режимі **Benchmark** аналізується швидкість та точність різних форматів експорту для YOLOv8. Бенчмарки надають інформацію про розмір експортованого формату, його метрики *mAP50-95* (для виявлення об'єктів, сегментації та позування) або *accuracy\_top5* (для класифікації) та час інференції в мілісекундах на зображення в різних форматах експорту, таких як ONNX, OpenVINO, TensorRT та інші. Ця інформація допомагає користувачам вибрати оптимальний формат експорту для їх конкретного випадку використання з урахуванням вимог до швидкості та точності.

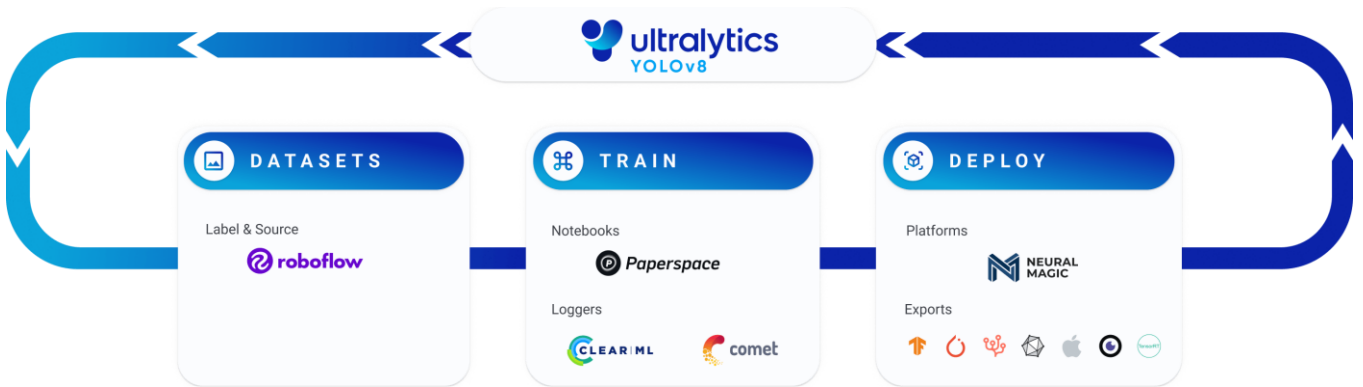


Рисунок 3.1 – Режими роботи YOLOv8

Фреймворк YOLOv8 може використовуватися для виконання завдань детекції, сегментації, класифікації та визначення пози. Кожне з цих завдань має різні цілі та сфери застосування.

Детекція є основним завданням YOLOv8. Вона полягає у виявленні об'єктів на зображенні або у відеокадрі та відмічанні їх обмежувальними рамками. Виявлені об'єкти класифікуються на різні категорії залежно від їхніх характеристик. YOLOv8 може виявляти кілька об'єктів на одному зображенні або у відеокадрі з високою точністю та швидкістю.

Сегментація - це завдання, яке полягає у поділі зображення на різні регіони відповідно до змісту зображення. Кожному регіону присвоюється мітка відповідно до його змісту. Це завдання корисне в таких застосуваннях, як сегментація зображень і медичне зображення. YOLOv8 використовує варіант архітектури U-Net для виконання сегментації.

Класифікація - це завдання, яке полягає у класифікації зображення на різні категорії. YOLOv8 може використовуватися для класифікації зображень залежно від їх змісту. Для виконання класифікації використовується варіант архітектури EfficientNet.

Визначення пози/ключових точок - це завдання, яке полягає у виявленні конкретних точок на зображенні або у відеокадрі. Ці точки називаються ключовими

та використовуються для відстеження руху або оцінки пози. YOLOv8 може виявляти ключові точки на зображенні або у відеокадрі з високою точністю та швидкістю.

### **3.3 Платформа машинного навчання PyTorch**

PyTorch, як відкрита платформа машинного навчання, була створена командою Facebook AI Research і швидко здобула популярність завдяки своїй адаптивності та простоті у використанні. Цей фреймворк відрізняється своєю гнучкістю і дозволяє дослідникам створювати складні нейронні мережі з легкістю.

Основною характеристикою PyTorch є динамічна природа його обчислювального графа, яка надає змогу міняти структуру мережі під час виконання. Це значно спрощує процес розробки та експериментування з нейронними мережами, оскільки він дозволяє більш гнучко працювати з вхідними даними та архітектурою мережі. Порівняно з іншими платформами, такими як TensorFlow, PyTorch пропонує більш інтуїтивно зрозумілий та інтерактивний підхід до побудови мережі.

PyTorch містить різноманітні інструменти та модулі, які спрощують процес створення, тренування та оцінки нейронних мереж. Його багатий набір вбудованих шарів, функцій втрати та алгоритмів оптимізації сприяє швидкій та ефективній розробці складних мережевих архітектур. Крім того, PyTorch легко інтегрується з іншими бібліотеками Python, що дозволяє використовувати широку гаму інструментів для обробки даних, візуалізації та оцінки моделей.

Особливістю PyTorch є його здатність до автоматичної диференціації, що дозволяє автоматично обчислювати градієнти для тензорів і параметрів мережі. Це полегшує зворотне поширення помилок та оновлення вагів моделі під час тренування, а також підвищує гнучкість у реалізації функцій втрати та методів оптимізації.

PyTorch також підтримує прискорення обчислень на GPU, особливо з використанням NVIDIA GPU. Це забезпечує ефективність у великомасштабних проєктах машинного навчання, де потрібна висока обчислювальна потужність.

PyTorch відзначається своєю активною спільнотою, яка постійно розширює його можливості, вносячи нові бібліотеки та інструменти. Це робить PyTorch важливим інструментом для дослідників та розробників, що працюють у сфері машинного навчання та комп'ютерного зору.

### **3.4 Функціональність по донавчанню мережі на власному наборі даних**

Окрім великої кількості вже зібраних наборів даних. Фреймворк YOLOv8 має можливість проводити навчання мережі на власно зібраному наборі даних.

Додавання нового набору даних до системи включає кілька кроків, щоб забезпечити його сумісність з існуючою інфраструктурою. Розглянемо необхідні кроки нижче.

Кроки для додавання нового набору даних:

- *Збір зображень*: Зібрати зображення, які належать до набору даних. Ці зображення можуть бути зібрані з різних джерел, таких як публічні бази даних або власна колекція.
- *Анотація зображень*: Позначити ці зображення обмежувальними рамками, сегментами або ключовими точками залежно від завдання.
- *Експорт анотацій*: Конвертувати ці анотації у формат файлу \*.txt, який підтримує Ultralytics.
- *Організація набору даних*: Розмістити ваш набір даних у відповідній структурі папок. Повинні бути каталоги верхнього рівня train/ і val/, і всередині кожного з них підкаталоги images/ і labels/.
- *Створення файлу data.yaml*: У кореневому каталозі вашого набору даних створіть файл data.yaml, що описує набір даних, класи та іншу необхідну інформацію.

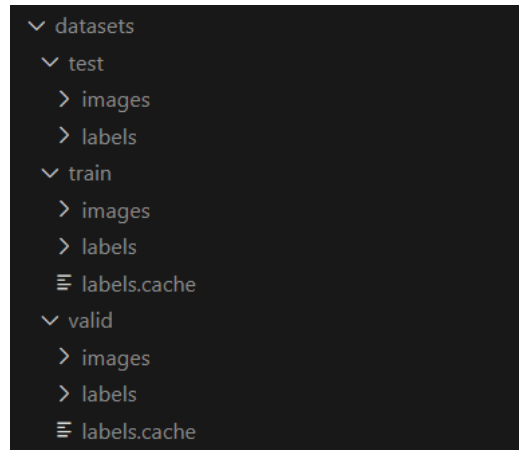


Рисунок 3.2 – Структура папок власного набору даних

### 3.5 Функціональність для вдосконалення архітектури мережі YOLOv8

Використовуючи пакет Ultralytics як фреймворк, а не застосунок, можна вносити зміни до архітектури моделі. Для цього розглянемо структуру файлів та класів фреймворку. Розглянемо детально Рисунок 3.3.

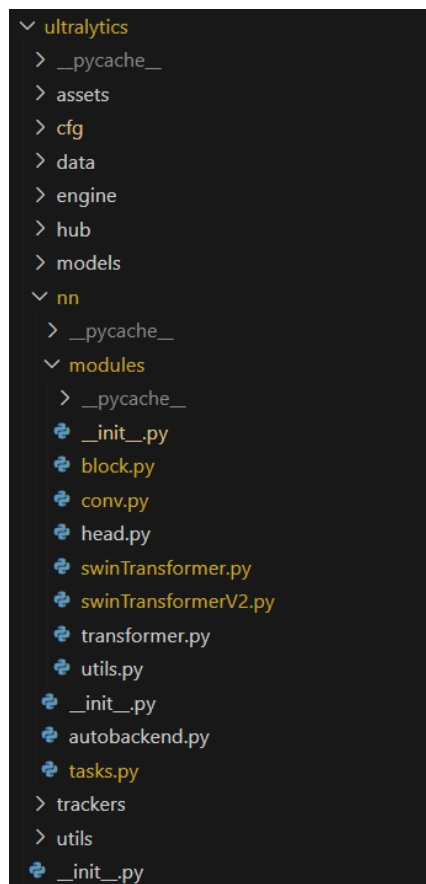


Рисунок 3.3 – Файлова структура фреймворку Ultralytics

Головна частина яка відповідає за модулі мережі YOLOv8 знаходиться у папці *ultralytics/nn/modules*. У цій папці знаходяться основні модулі, такі як *backbone*, *neck* та *head*. Файл *conv.py* містить в собі класи які відповідають за методи згортки.

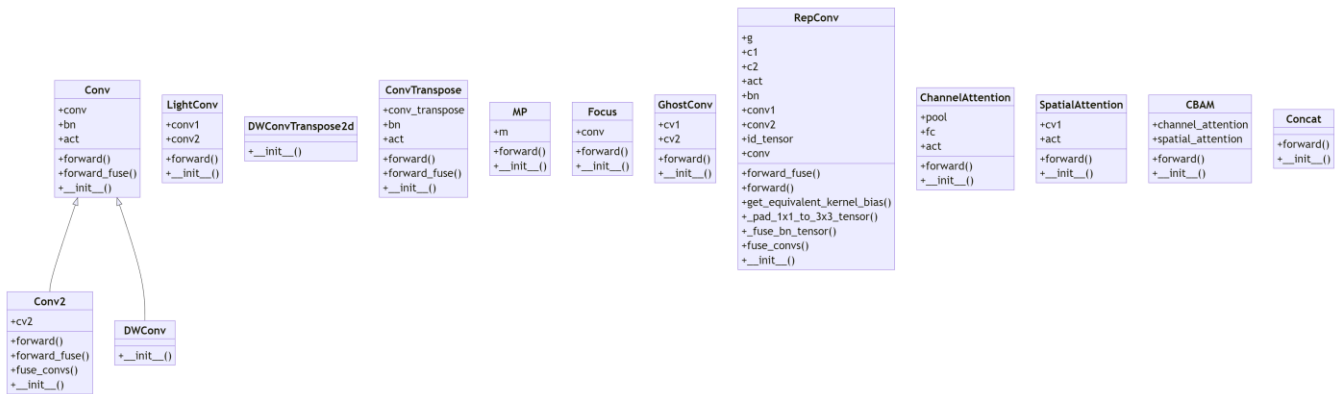


Рисунок 3.4 – Діаграма класів файлу *conv.py*

Класи детекції, сегментації, класифікації та визначення пози/ключових точок знаходяться у файлі *head.py*. На Рисунок 3.6 зображено діаграму класів модулю *head*.

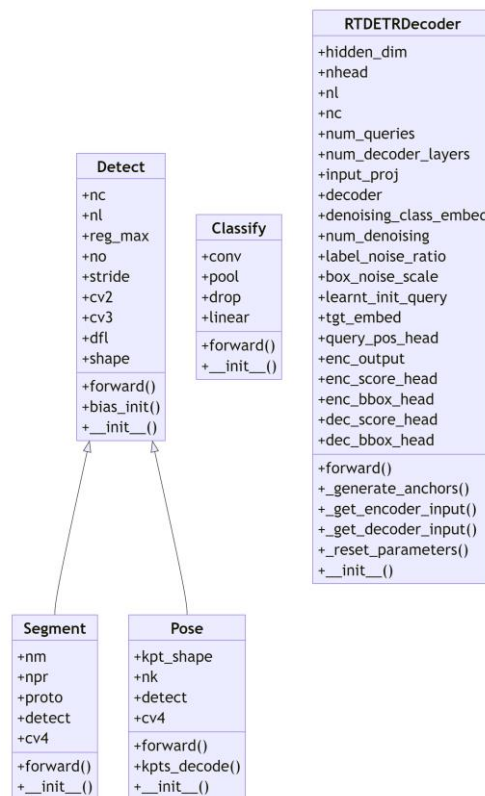


Рисунок 3.6 – Діаграма класів файлу *head.py*

У файлі `block.py` знаходяться класи модулю `backbone`. Такі як `Bottleneck`, `SPPF`, `C1`, `C2` та `C3`. Діаграму класів наведено на Рисунку 3.7.

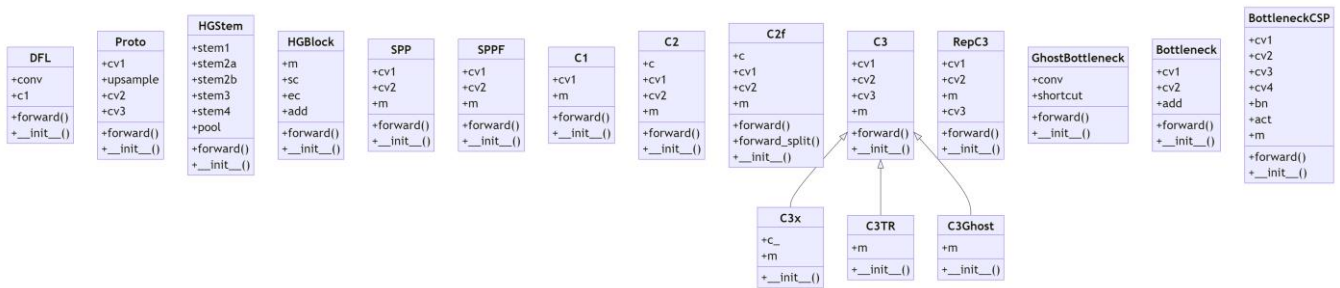


Рисунок 3.7 – Діаграма класів файлу `block.py`

### Висновки до розділу 3

У цьому розділі розглянуто ключові етапи розробки інформаційної системи для виявлення багатороторних БПЛА, заснованої на нейронній мережі YOLOv8. Система інтегрує передові підходи з галузей глибокого навчання та комп'ютерного зору, зосереджуючись на точному виявленні та класифікації об'єктів.

Було обрано Python як засіб розробки завдяки його гнучкості, об'єктно-орієнтованому програмуванню та широкій підтримці у спільноті машинного навчання.

У рамках дослідження було детально проаналізовано структурні елементи фреймворку Ultralytics, а також його ключові модулі, включаючи `backbone`, `neck` та `head`. Даний аналіз забезпечив глибоке розуміння функціонування та внутрішньої структури моделі, що є критично важливим для виконання поставленого завдання.

Розгляд архітектури мережі YOLOv8 виявив її потенціал у виявленні та класифікації об'єктів з високою точністю та швидкістю. Зокрема, було визначено, як модель може бути адаптована та оптимізована для специфічних потреб виявлення БПЛА. Дослідження охопило також питання тренування та оцінювання моделі, її можливості для вдосконалення та інтеграції з іншими інструментами та платформами.

Завдяки цьому аналізу було сформовано міцну основу для подальшої розробки системи, яка здатна ефективно виявляти та класифікувати БПЛА. Розуміння ключових аспектів архітектури фреймворку Ultralytics та її використання у контексті поставленні задачі виявлення БПЛА відкриває нові можливості для розширення можливостей системи та підвищення її ефективності.

## РОЗДІЛ 4

### Експериментальні результати виявлення багатороторних БПЛА запропонованим методом

#### 4.1 Опис застосування для проведення експериментів

Ключову роль у проведенні експериментів займає спеціалізоване програмне забезпечення, розроблене для тестування запропонованого методу та призначене аналізу та виявлення БПЛА за допомогою нейронної мережі YOLOv8. Дане програмне рішення було спеціально адаптоване для вирішення завдань класифікації та виявлення об'єктів у великих масивах візуальних даних.

Функціональність застосування дозволяє реалізувати, наведені далі, можливості.

- *Обробка Великих Даних*: Програма здатна ефективно обробляти великі обсяги зображень та відео, що є основою для точного виявлення об'єктів.
- *Інтеграція з YOLOv8*: Центральним елементом програми є інтеграція з нейронною мережею YOLOv8, що дозволяє використовувати передові технології машинного навчання для аналізу зображень.
- *Гнучкість налаштувань*: Програма надає можливість тонкого налаштування параметрів моделі та обробки даних, що дозволяє оптимізувати процес виявлення залежно від конкретних потреб дослідження.
- *Візуалізація результатів*: Після обробки даних програма забезпечує візуалізацію результатів, що включає ідентифікацію та класифікацію об'єктів, а також надає детальні звіти про виявлені об'єкти.
- *Сумісність та інтеграція*: Програма сумісна з різними операційними системами та може бути інтегрована з іншими інструментами та платформами для розширення її можливостей.

Для обробки результатів тренування мережі було використано платформу Comet ML. Comet ML є важливою частиною процесу розробки та використання

неймереж, зокрема YOLOv8. Це платформа для відстеження, порівняння, пояснення та оптимізації моделей машинного навчання та експериментів. Вона дозволяє реєструвати метрики, параметри, медіа та інше під час навчання моделі, а також моніторити експерименти через естетично приємний веб-інтерфейс. Comet ML підвищує прозорість та відтворюваність, і сприяє розробці продуктивних моделей.



Рисунок 4.1 – Інтерфейс платформи Comet ML

Інтеграція YOLOv8 з Comet ML відкриває широкий спектр переваг. Це включає в себе спрощення управління експериментами, отримання реальних відомостей для швидких коригувань, гнучкі та адаптивні варіанти реєстрації, а також можливість реєструвати експерименти в автономному режимі, коли доступ до Інтернету обмежений. Ця інтеграція дає можливість приймати рішення, засновані на даних, аналізувати показники продуктивності та досягати виняткових результатів.

Після запуску коду навчання Comet ML автоматично створює експеримент у вашому робочому просторі Comet для відстеження цього процесу. Після цього надається посилання для перегляду детального реєстрування процесу навчання моделі YOLOv8.

Comet ML має потужний інтерфейс для відстеження та аналізу експериментів. Цей інтерфейс забезпечує централізоване середовище для контролю, порівняння та оцінки ефективності різних моделей та підходів до навчання.

Сторінка експерименту на платформі Comet ML містить декілька вкладок, кожна з яких пропонує різні інструменти для аналізу та оцінки моделі машинного навчання. Вона є детальним інтерфейсом, що надає всеосяжний огляд машинного навчання. Ось детальний опис вкладок на сторінці експерименту Comet ML:

- **Panels:** дозволяє користувачам створювати індивідуальні візуалізації та панелі управління, використовуючи дані з їхніх експериментів. Користувачі можуть налаштовувати ці панелі для відображення метрик, найбільш відповідних їхнім конкретним вимогам.

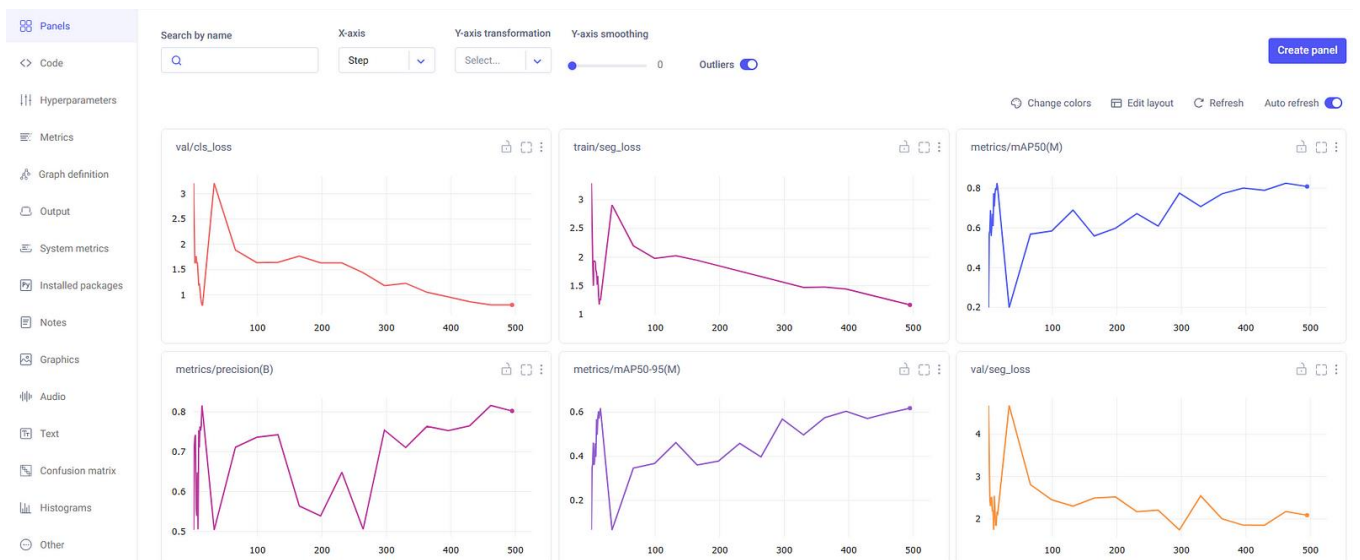


Рисунок 4.2 – Візуальний вигляд вкладки Panels

- **Metrics:** демонструє різні метрики, що відстежуються під час експерименту. Ці метрики, такі як втрати та точність, візуалізуються у графіках, які оновлюються в реальному часі, надаючи інформацію про продуктивність моделі протягом процесу навчання.

– Confusion Matrix: надає інтерактивний спосіб оцінки точності класифікації моделі. Вона докладно показує правильні та неправильні прогнози, дозволяючи зрозуміти сильні та слабкі сторони моделі.

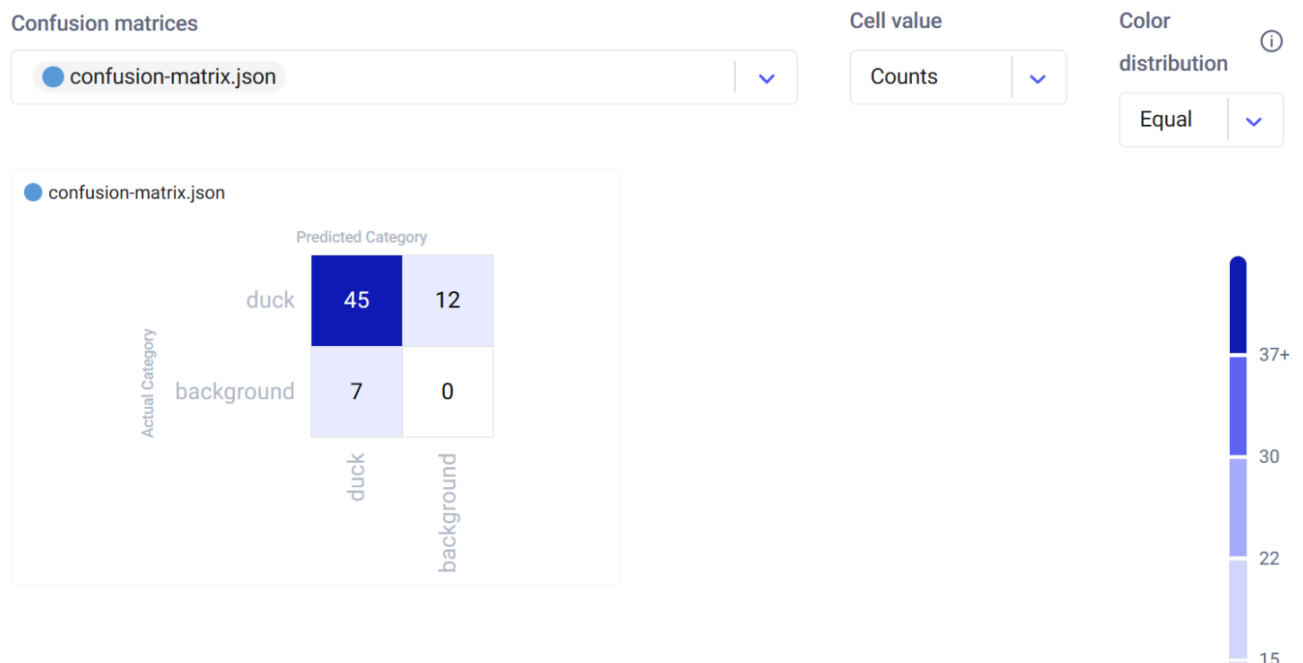


Рисунок 4.3 – Візуальний вигляд Confusion Matrix

– Code: містить код, який був виконаний для експерименту. Це цінний ресурс для перегляду і розуміння точних процедур і параметрів, використаних у експерименті, забезпечуючи відтворюваність і прозорість у процесі моделювання.

– Graphs: тут можна знайти візуалізації обчислювального графа моделі. Це візуальне представлення допомагає зрозуміти архітектуру моделі та потік даних через мережу.

Рисунок 4.4 – Вкладка Graph definition

– System Metrics: Ця вкладка відстежує та відображає системні метрики, такі як використання CPU і GPU, дозволяючи користувачам оцінювати використані обчислювальні ресурси під час експерименту.

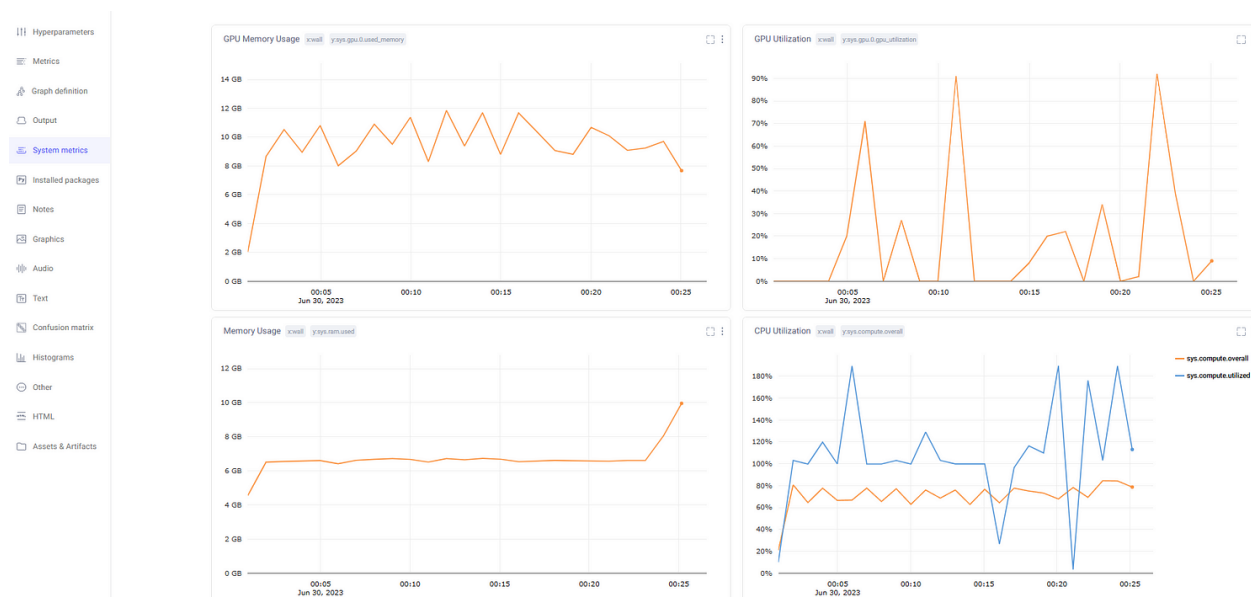


Рисунок 4.5 – Приклад графіків використання CPU і GPU



Рисунок 4.6 – Візуальний вигляд вкладки System Metrics

– HTML: користувачі можуть додавати HTML-базовані нотатки на вкладку "HTML" під час виконання експерименту, додаючи важливі анотації та інсайти за допомогою методів `Experiment.log_html()` і `Experiment.log_html_url()` з Python SDK.

## 4.2 Опис набору даних для тестування та параметрів навчання

Для оцінки продуктивності моделі було використано загальнодоступний набір даних UAV TIV-Net [42]. Набір даних UAV TIV-Net складається з 2850 зображень, зображення містять різні типи БПЛА, включаючи багатороторні БПЛА та БПЛА з фіксованим крилом. Зображення були зняті фіксованою камерою на землі на відстані близько 500 м від дронів, а роздільна здатність зібраних зображень становить  $1920 \times 1080$  пікселів. Ці зображення включають різноманітні сценарії польотів БПЛА на низькій висоті (такі як небо, дерева, будівлі та інше), при цьому вони покривають ситуації польотів в різні часи дня та за різних метеорологічних умов (рис. 4.7).



Рисунок 4.7 – Приклад зображень з набору даних UAV TIB-Net [42]

Для тренування встановлено ліміт у 150 епох, при цьому перші три епохи використовуються для початкового "розігріву" моделі. Оптимізація процесу навчання здійснюється за допомогою стратегії SGD (стохастичний градієнтний спуск) з початковою швидкістю навчання, встановленою на рівні 0.01.

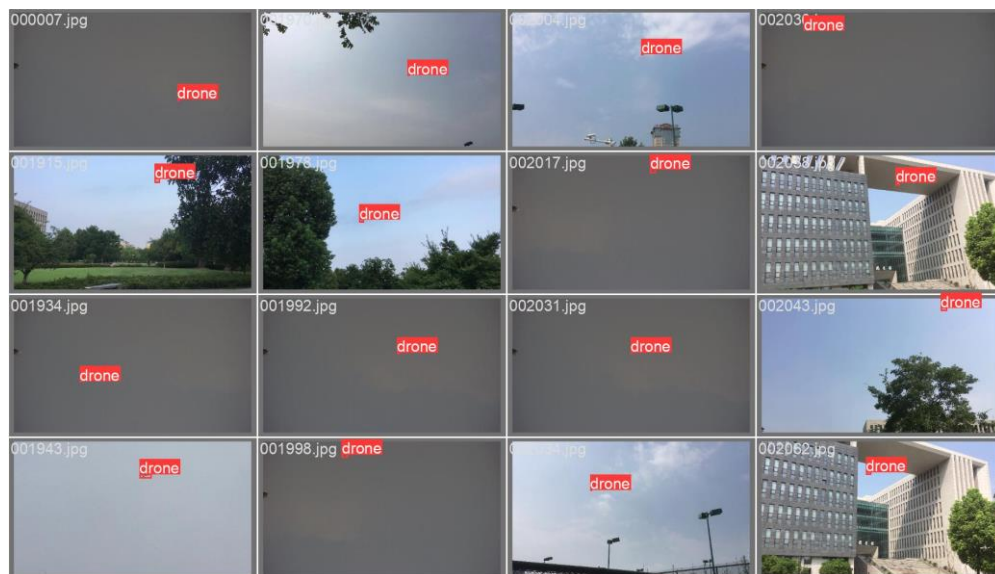


Рисунок 4.8 – Приклад анотованих зображень набору даних UAV TIB-Net.

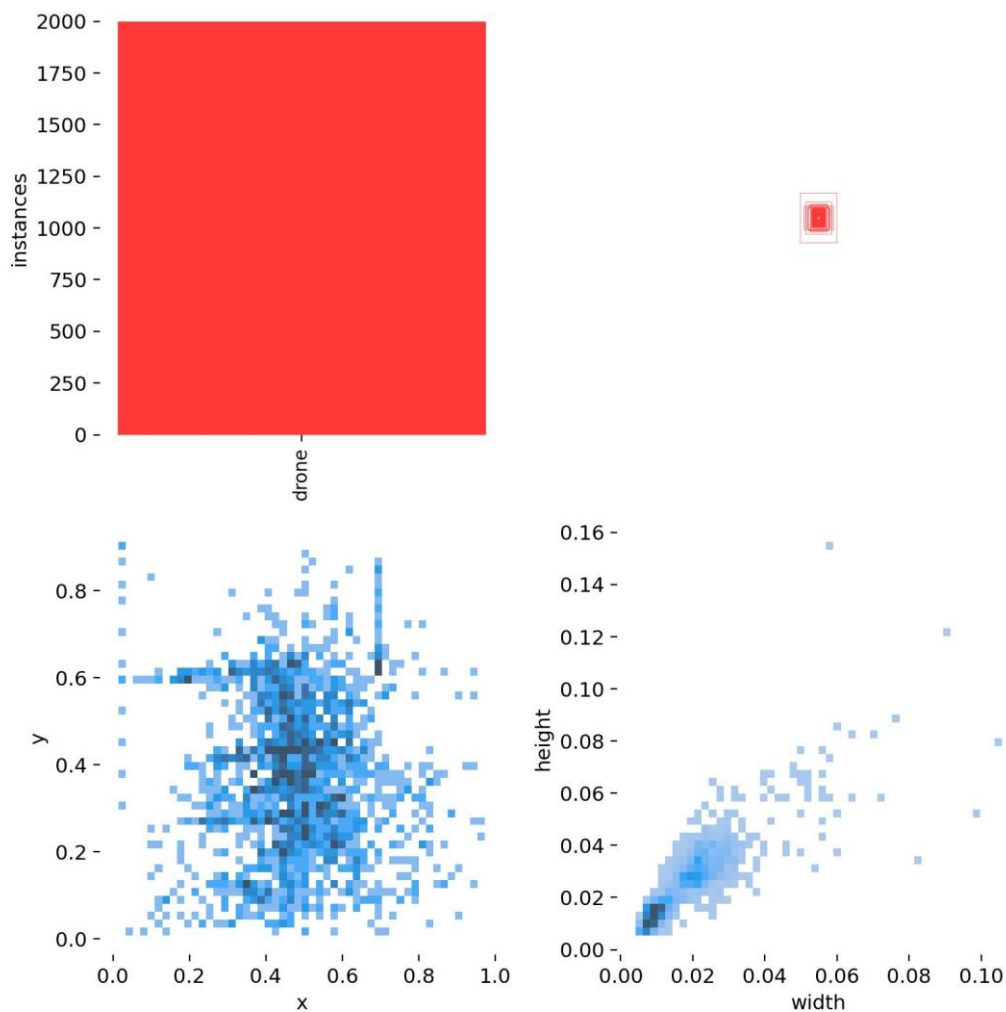


Рисунок 4.9 – Співвідношення розміру дронів до розміру зображень. Більш темні зони означають більшу концентрацію зображень.

Для тренування встановлено ліміт у 150 епох, при цьому перші три епохи використовуються для початкового "розігріву" моделі. Оптимізація процесу навчання здійснюється за допомогою стратегії SGD (стохастичний градієнтний спуск) з початковою швидкістю навчання, встановленою на рівні 0.01. З огляду на високу частоту малорозмірних об'єктів на зображеннях та необхідність знаходження оптимального балансу між швидкістю обробки в реальному часі та точністю виявлення, розмір вхідних зразків було нормалізовано до 640×640 пікселів. Такий підхід дозволяє використовувати модель з мінімальними втратами важливої інформації на зображеннях. Для забезпечення об'єктивності порівняльного аналізу

продуктивності різних моделей у експериментах, переднавчена модель не буде використовуватися. Всі процеси тренування налаштовані відповідно, з ключовими параметрами, які наведено в Таблиці 4.1.

Таблиця 4.1 – Таблиця параметрів.

Параметри	Значення
Epochs	150
Warmup-epochs	3
Warmup-momentum	0.8
Batch Size	8
Imgsize	640
Initial Learning Rate	0.01
Final Learning Rate	0.01
Patience	50
Optimizer	SGD
NMSIoU	0.7
Momentum	0.937
Mask-ratio	4
Weight-Decay	0.0005

### 4.3 Власний набір даних для донавчання нейромережі

Донавчання мережі проводилося з використанням власного набору даних. Було зібрано загалом 10000 зображень з різноманітних датасетів. Ці зображення містять різні моделі БПЛА у різноманітних сценах та періодах доби.



Рисунок 4.10 – Приклади зображень із власного набору даних

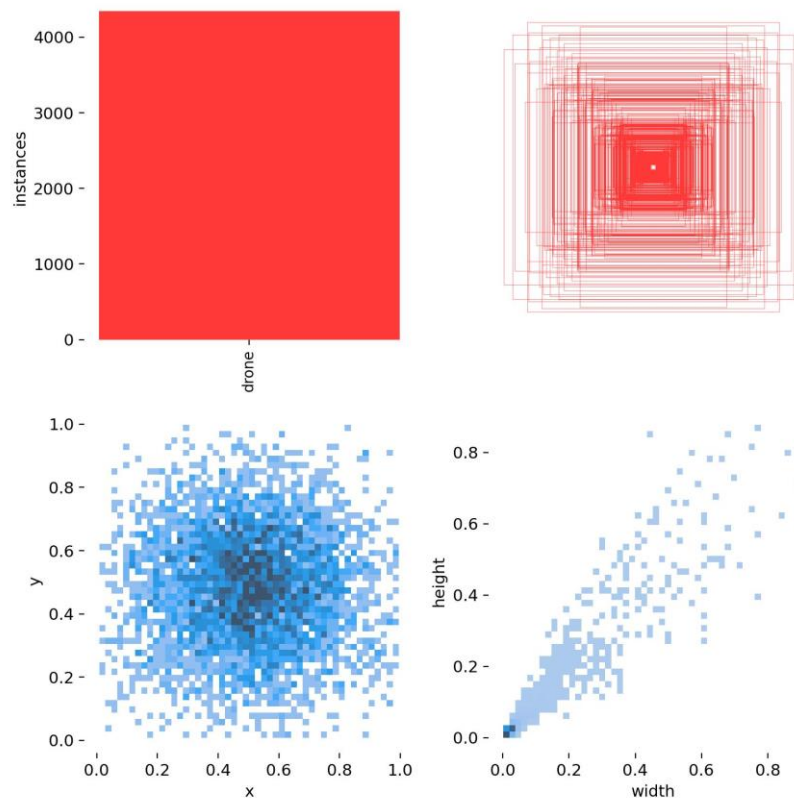


Рисунок 4.11 – Співвідношення розмірів БПЛА до розміру зображень

На основі аналізу Рисунку 4.11, можна зробити висновок, що пропорція присутності більшості БПЛА на аналізованих зображеннях є досить незначною,

складаючи менше одного відсотка від загальної площі кадру. Це вказує на важливість розробки ефективних методів для виявлення та класифікації таких малорозмірних об'єктів в середовищі великих наборів даних.

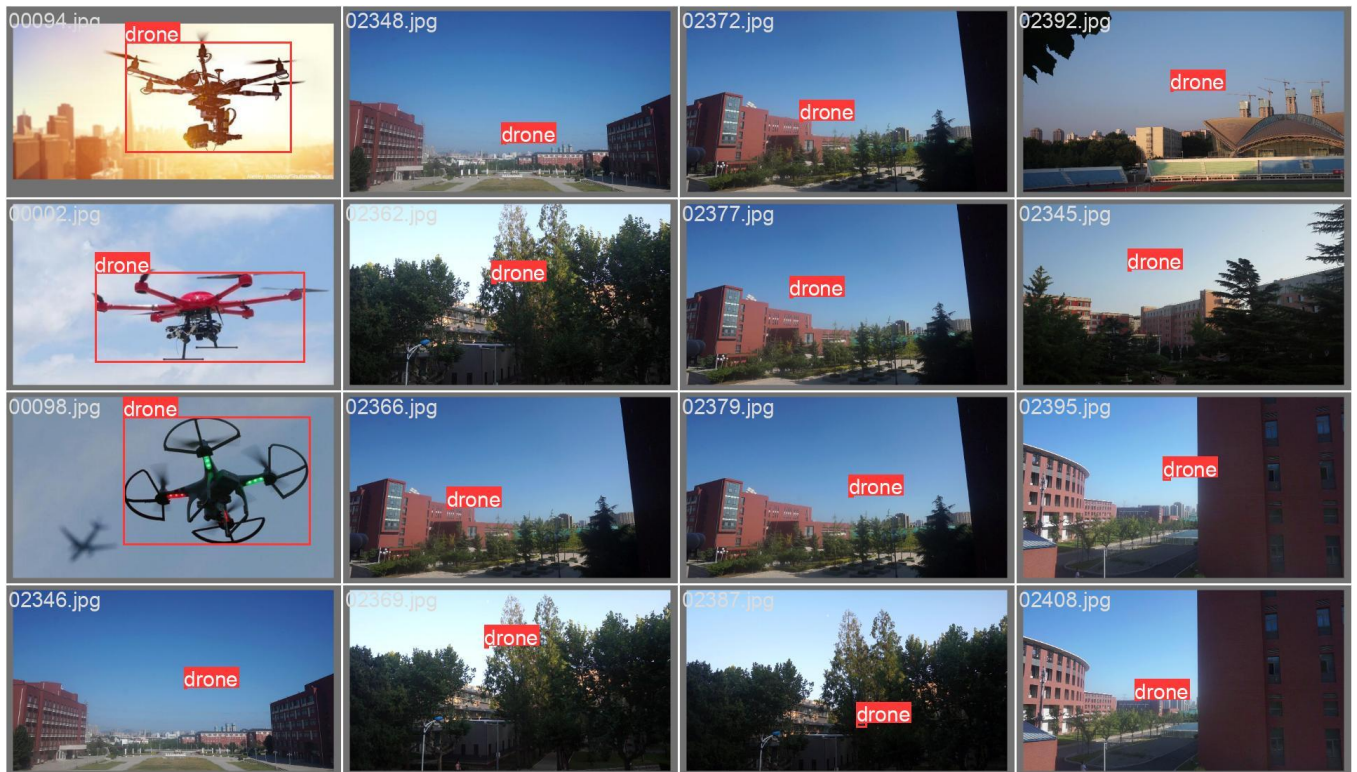


Рисунок 4.12 – Приклад анотованих зображень власного набору даних.

Як можна помітити з Рисунку 4.9 порівняно із набором TIV-Net у нашому наборі даних набагато більше зображень різноманітних розмірів, а також є достатня концентрація у зоні малих об'єктів.

#### 4.4 Вдосконалення архітектури мережі YOLOv8

Для вдосконалення архітектури YOLOv8. Роботу було розбито на декілька етапів.

– *Етап 1:* додавання класу C3STR який буде заміною для звичного модулю Conv у частині Backbone. Для цього створимо новий файл SwinTransformer.py на рівні із модулями head, block та conv. Потім до файлу block.py

додамо клас C3STR. На Рисунку 4.13 зображено діаграму класів SwinTransformer.py та діаграму класу C3STR.

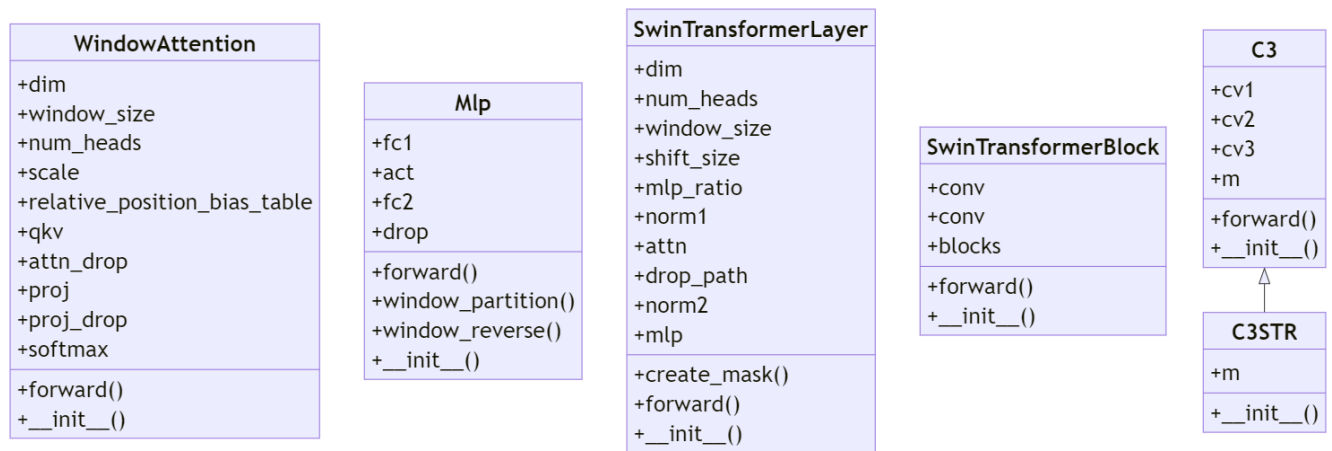


Рисунок 4.13 – Діаграма класів SwinTransformer

– *Етап 2*: на цьому етапі необхідно до модулю head додати механізм уваги GAM. Для цього вносимо зміни до файлу `conv.py`.

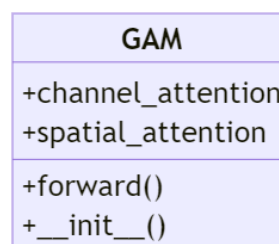


Рисунок 4.14 - Діаграма класу GAM

– *Етап 3*: вносимо зміни до основної конфігурації моделі. Файл конфігурації має назву `yolov8.yaml`. Саме у цьому файлі прописані всі інструкції по послідовності використання тих чи інших модулів. У ньому було додано шар детектування P2 та видалено шар детекції великих об'єктів P5.

В результаті змін у архітектурі YOLOv8 була отримана поліпшена версія моделі, яка оптимізована для більш ефективного виявлення багатороторних БПЛА. Нова конфігурація забезпечує підвищену чутливість до малих об'єктів та більшу адаптивність до різноманітних сценаріїв виявлення, завдяки інтеграції механізму



*Модель 1* – базова модель YOLOv8s, що слугує як стартова точка.

*Модель 2* – Розширення базової моделі за рахунок інтеграції додаткового детекційного шару P2 у модуль head (+Tiny-Head).

*Модель 3* – На основі моделі 2, з вилученням детекційного шару, що відповідає за виявлення великих об'єктів (-Large-Head), зберігаючи доданий шар P2.

*Модель 4* – Подальше удосконалення *Моделі 3* шляхом додавання Swin Transformer (+C3SPR), зберігаючи попередні зміни.

*Модель 5* – Вдосконалення *Моделі 4*, для цього до неї додамо GAM.

*Модель 6* – Найбільш комплексне удосконалення, яке включає додавання Swin Transformer та GAM до моделі з детекційним шаром P2, але без детекційного шару для великих цілей.

Зміни показників оцінювання для цих шести моделей були кількісно досліджені, і оптимальні результати для кожного показника оцінювання були виділені. Експериментальні результати моделей на наборі даних TIB-Net показані в Таблиці 4.2.

Таблиця 4.2 – Результати експериментів на наборі даних TIB-Net.

Компоненти	1	2	3	4	5	6
+Tiny-Head		+	+	+	+	+
-Large-Head			+	+	+	+
+C3SPR				+		+
+GAM					+	+
P	81.4%	87.9%	87.5%	89.5%	88.2%	90.4%
R	78.1%	86.5%	86.5%	88.1%	86.2%	88.3%
mAP	86.1%	91.2%	90.3%	91.7%	90.6%	92.1%
Parameters/million	11.126	10.852	3.527	6.83	8.09	7.511
Model Size/MB	21.9	22.1	13.1	13.9	13.4	14.8
FPS/f.s-1	285	196	226	199	213	191

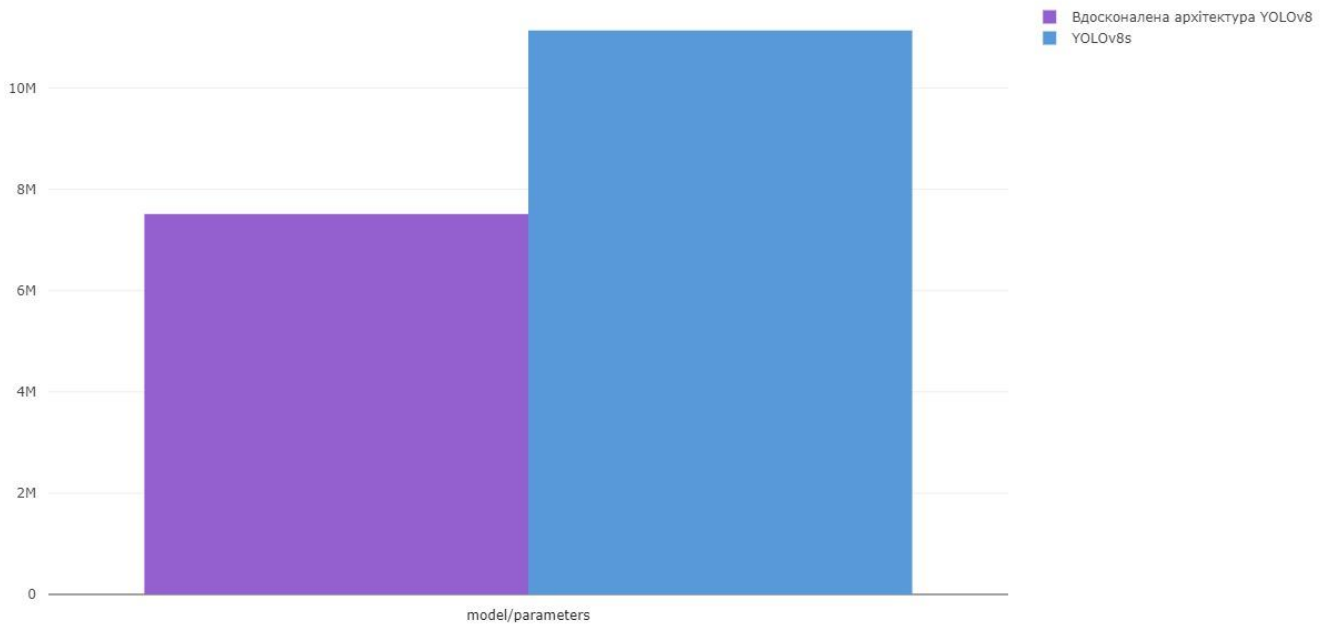


Рисунок 4.16 – Порівняння кількості параметрів моделі.

Додавання детекційного шару P2 у модуль head покращило модель на 6.5%, 8.4% та 5.1% відповідно для P, R та mAP, це свідчить про те, що додавання такого модулю ефективно підвищує здатність моделі виявляти малі цілі. Водночас, після видалення шару детектування великих цілей кількість параметрів зменшилась на 70.2%, а розмір моделі - на 67%, при цьому R залишився незмінним, P зменшився на 0.4%, а mAP - на 0.9%, це свідчить про те, що шар детектування з низькою роздільною здатністю мало впливає на виявлення малих цілей БПЛА.

Експериментальні результати покращення моделей 3, 4, 5 та 6 показують, що додавання Swin Transformer дає кращий вплив на показник відгуку R моделі, це означає, що заміна модулю Conv на Swin Transformer, у модулі backbone мережі, дозволяє краще зберігати характеристики мікроцілей і знижує ймовірність пропуску виявлення мікроцілей; додавання GAM покращило точність P моделі. Тобто, додавання модулю уваги GAM у модуль neck мережі позитивно впливає на злиття характеристик мережі і знижує ймовірність помилкового виявлення мережею. Коли були додані як Swin Transformer, так і GAM, показники P, R і mAP покращились, хоча кількість параметрів та розмір моделі трохи збільшилися.

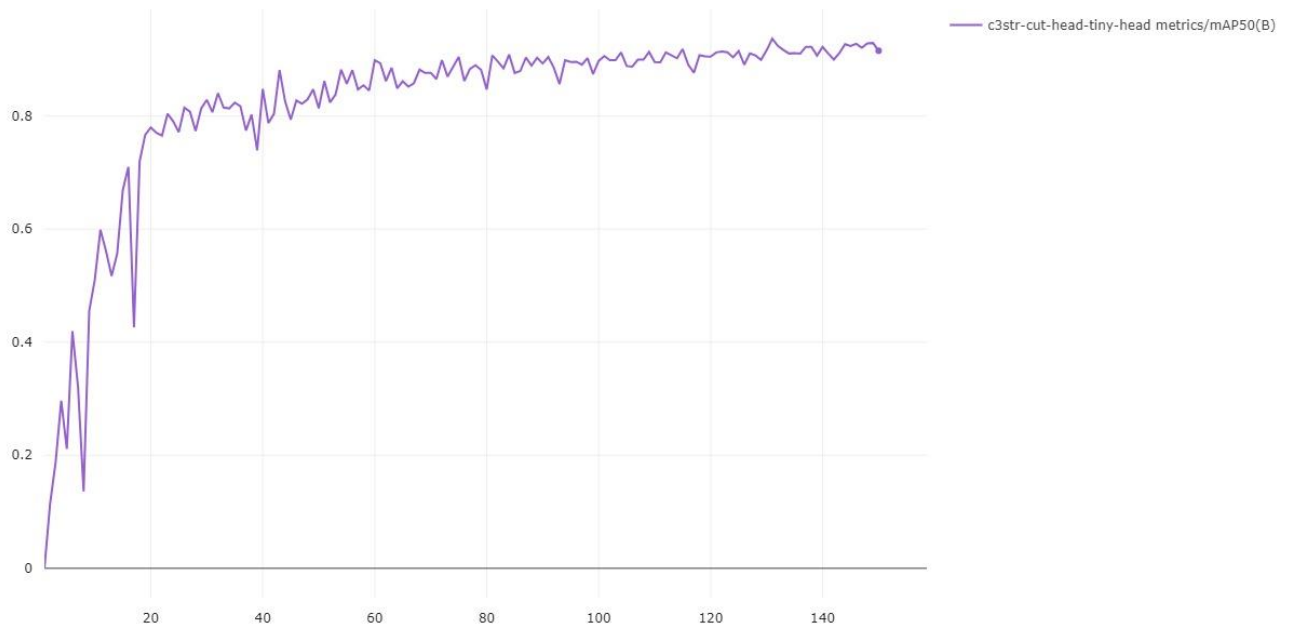


Рисунок 4.17 – Динаміка зміни mAP50(B) відносно епохи навчання (+C3SPR, +Tiny-Head, -Large-Head).

Внесення C3SPR і Tiny-Head, при видаленні Large-Head, призводить до більш швидкого зростання mAP50, особливо на ранніх стадіях тренування. Це вказує на те, що модифікації покращують здатність мережі ефективно виявляти малі об'єкти, як-от БПЛА, з самого початку навчання.

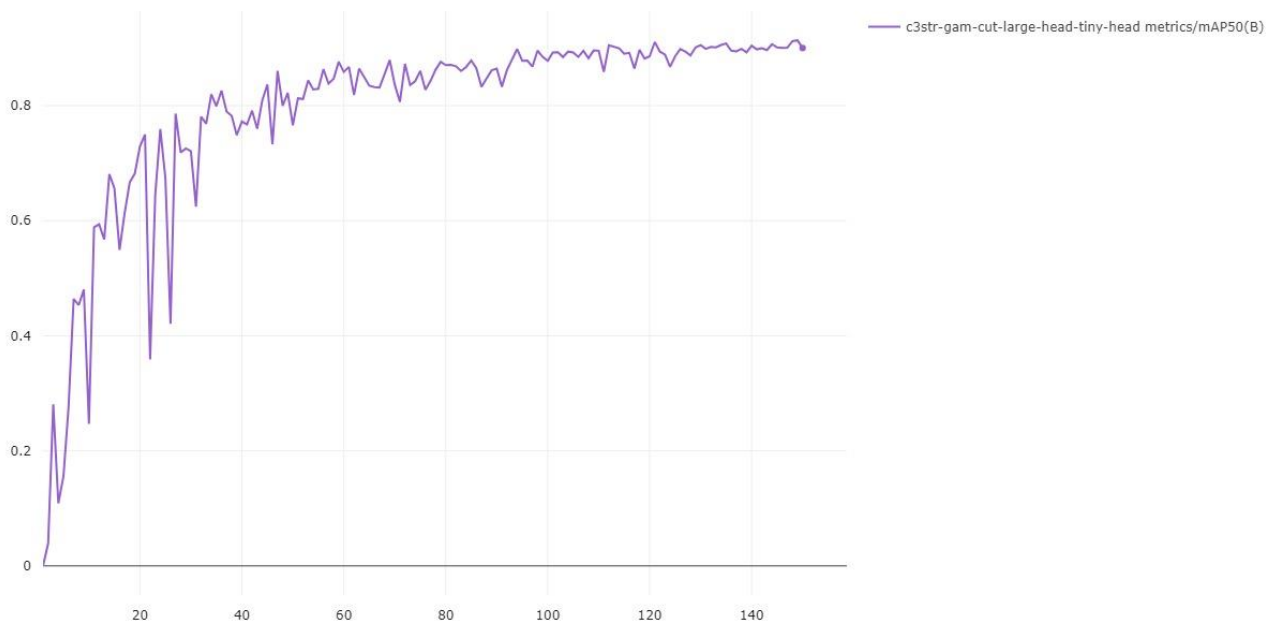


Рисунок 4.18 – Динаміка зміни mAP50(B) відносно епохи навчання (+C3SPR, GAM, +Tiny-Head, -Large-Head).

Наприкінці процесу тренування, можна помітити стабілізацію mAP50, це вказує на досягнення моделлю своєї оптимальної точності.

Цей графік є важливим інструментом для оцінки та аналізу ефективності внесених змін у моделі YOLOv8, демонструючи, як конкретні модифікації впливають на загальну продуктивність моделі в процесі її навчання.

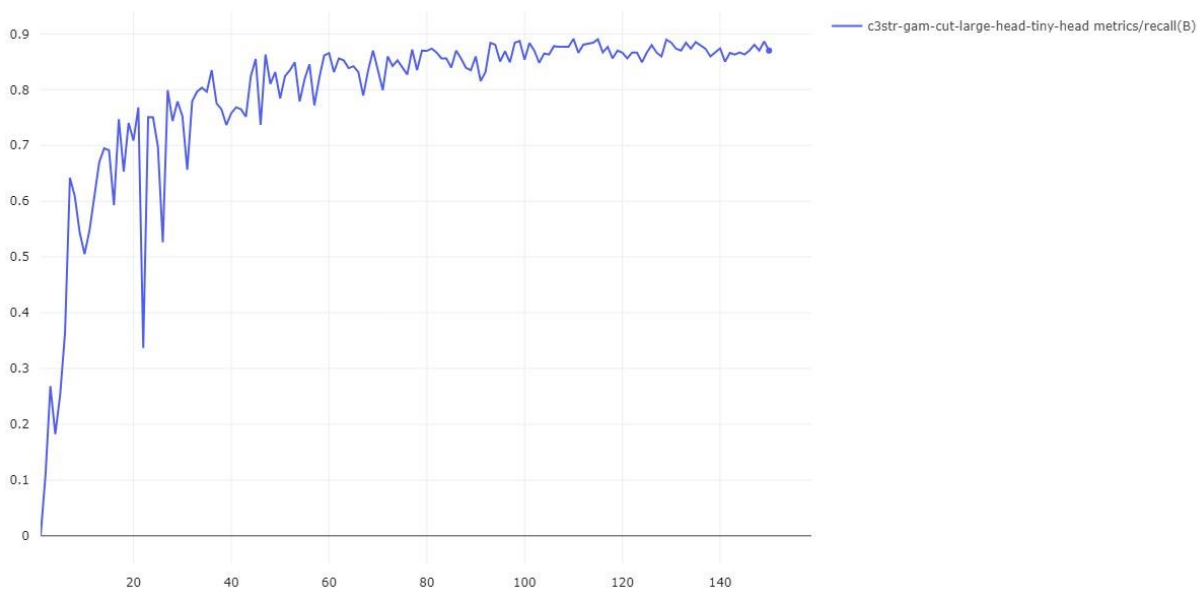


Рисунок 4.19 – Динаміка зміни R(B) відносно епохи навчання (+C3SPR, GAM, +Tiny-Head, -Large-Head).

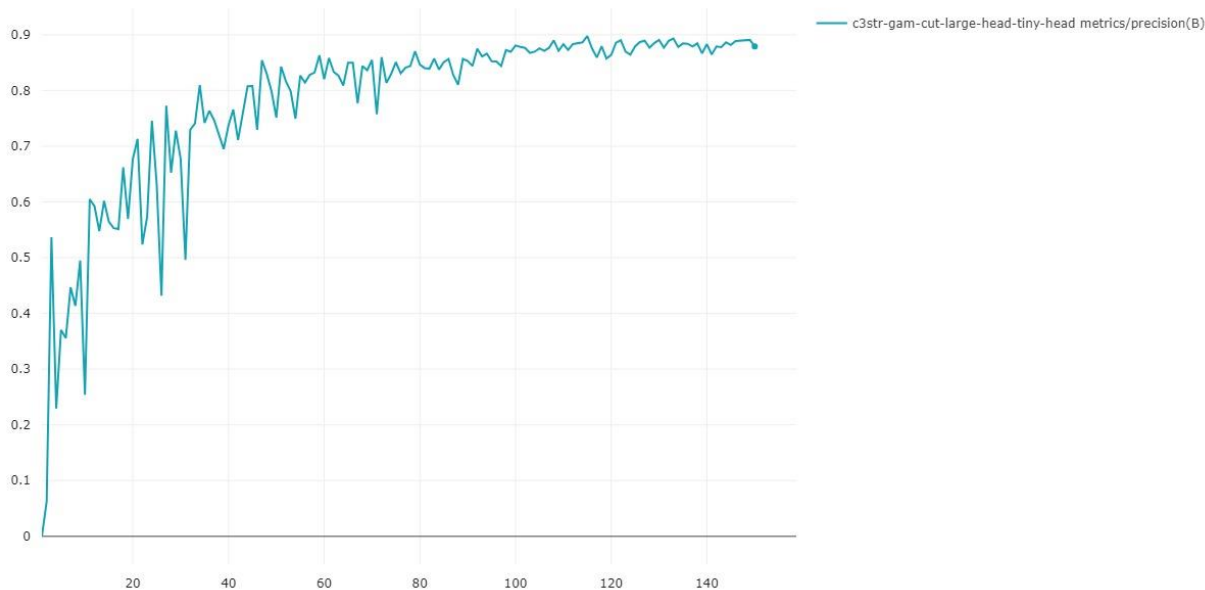


Рисунок 4.20 – Динаміка зміни P(B) відносно епохи навчання (+C3SPR, GAM, +Tiny-Head, -Large-Head).

Порівнюючи експериментальні результати покращеної *Моделі 6* та *Моделі 1* (тобто базової моделі), можемо спостерігати, що наша модель значно покращила показники P, R, mAP, кількість параметрів та розмір моделі порівняно з базовою моделлю. У порівнянні з початковою базовою моделлю YOLOv8, нова модель покращує продуктивність на 9%, 10,2% та 6% за показниками P, R та mAP відповідно. Кількість параметрів та розмір моделі зменшились на 32,5% та 32,4% відповідно, що підтверджує ефективність та практичність покращеної моделі.

#### **4.6 Порівняння отриманих результатів тестування на власному наборі даних**

Тренування мережі було здійснено на прикладі стандартної моделі YOLOv8s та модифікованої моделі з усіма рекомендованими поліпшеннями. Цей процес включав адаптацію моделей до специфіки власно зібраного набору даних, дозволяючи провести глибокий аналіз їх продуктивності. Такий підхід забезпечив можливість порівняти ефективність стандартної моделі YOLOv8s з моделлю, що була оптимізована згідно з нашими рекомендаціями, і оцінити вплив запропонованих вдосконалень на загальну продуктивність системи виявлення БПЛА.

Під час навчання на власному наборі даних, стандартна модель YOLOv8s продемонструвала наступні результати: точність (Precision, P) - 92%, повнота (Recall, R) - 77.2% та середня точність по всіх класах (mean Average Precision, mAP) - 85.3%. В той час як вдосконалена модель показала значне поліпшення: точність - 95.8%, повнота - 85.5% та mAP - 91.8%. Результати зображені на Рисунках 4.21, 4.22 та 4.23.

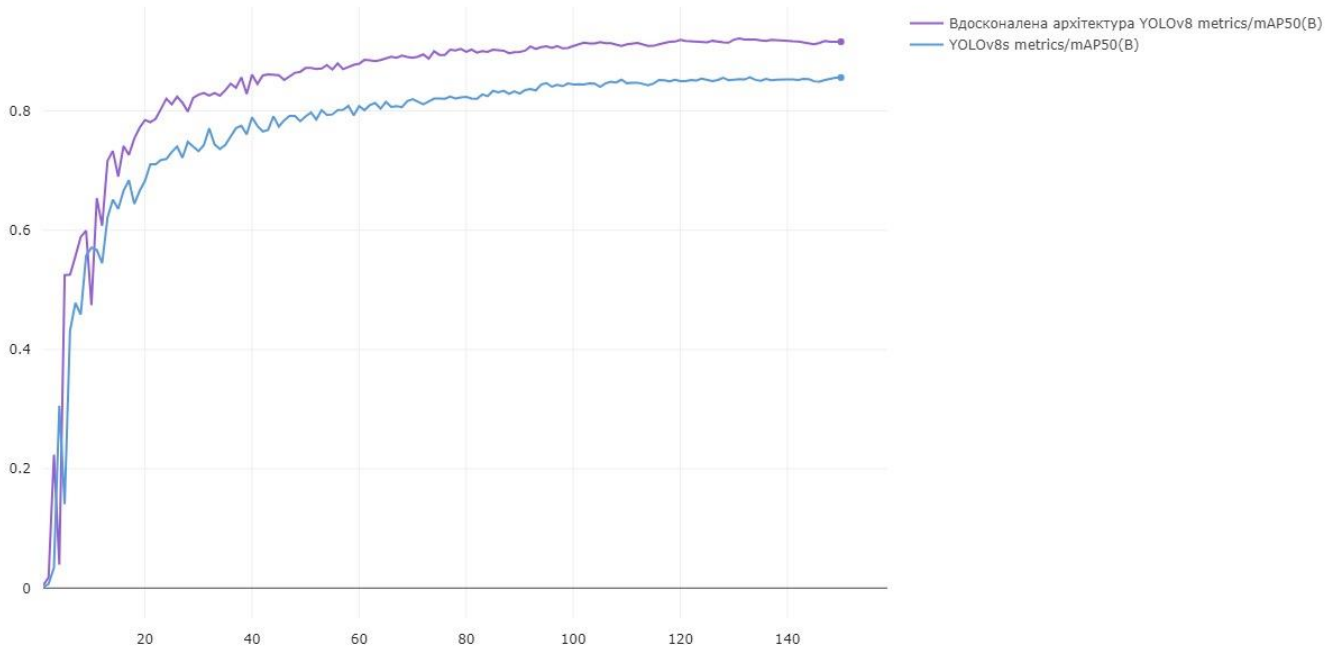


Рисунок 4.21 – Порівняння значень середньої точності (mAP).

На Рисунку 4.21 зображено дві криві, що ілюструють динаміку зміни показника mAP50(B) відносно кількості епох навчання для двох архітектур нейронної мережі YOLOv8: стандартної та вдосконаленої. mAP50 являє собою середню точність виявлення з порогом перекриття 50%.

Синя крива відображає результати для стандартної моделі YOLOv8, тоді як фіолетова показує результати для моделі з внесеними вдосконаленнями. Можна побачити, що обидві криві демонструють швидке зростання mAP50(B) на ранніх етапах тренування, що вказує на ефективне навчання моделі. Зростання сповільнюється з часом, зближуючись до певного насичення, що є типовим для процесу тренування нейронних мереж, коли модель досягає свого оптимального стану.

На графіку можна також помітити, що фіолетова крива вдосконаленої моделі знаходиться вище синьої в більшості точок, що свідчить про те, що модель з вдосконаленнями показує кращу загальну ефективність у виявленні класу В протягом усього процесу навчання. Це вказує на те, що додавання класу C3STR та

інших змін позитивно впливає на здатність моделі виявляти об'єкти, що мають важливе значення при виявленні малих об'єктів, які можуть бути складно розпізнати.

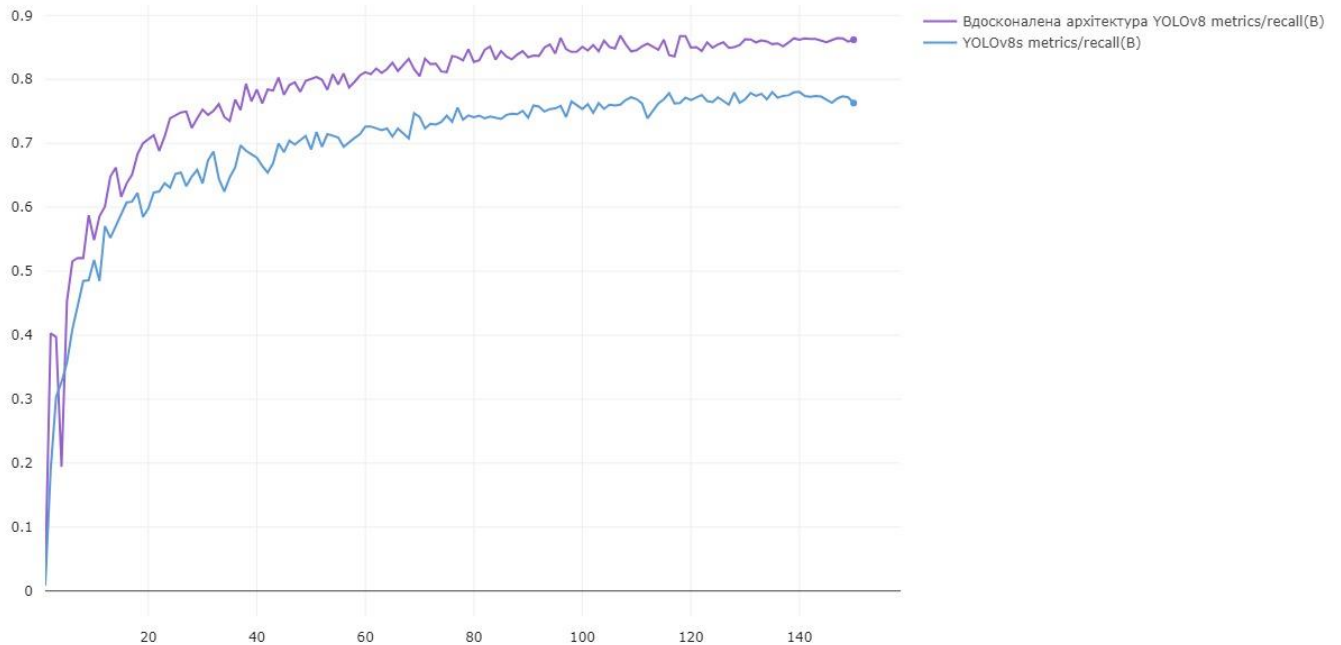


Рисунок 4.22 – Порівняння значень повноти (R).

Показник повноти оцінює здатність моделі виявляти всі релевантні зразки в датасеті, тобто він вказує на частку правильно ідентифікованих об'єктів класу В серед усіх об'єктів цього класу, що мали бути виявлені.

За допомогою двох кривих: фіолетової для вдосконаленої моделі та синьої для стандартної, можна зробити декілька важливих спостережень:

Обидві моделі демонструють стрімке збільшення відгуку на початкових етапах навчання, що свідчить про їх спроможність до швидкого навчання і здатності до розпізнавання класу В.

Фіолетова крива показує дещо вищий показник відгуку в порівнянні з синьою на більшості етапів, що може свідчити про підвищену здатність вдосконаленої моделі виявляти об'єкти класу В.

Значення відгуку для обох моделей стабілізуються близько позначки 0.8, і, подібно до mAP50(B), після певної кількості епох навчання, додаткові епохи не приносять значного покращення.

Це може вказувати на обмеження в даних або потенційну перенавченість моделі, коли додаткове тренування не веде до покращення загальної здатності моделі розпізнавати нові зразки. Важливо зазначити, що показник відгуку важливий для застосувань, де критично важливо не пропустити жодного релевантного об'єкта, наприклад, в системах безпеки або моніторингу.

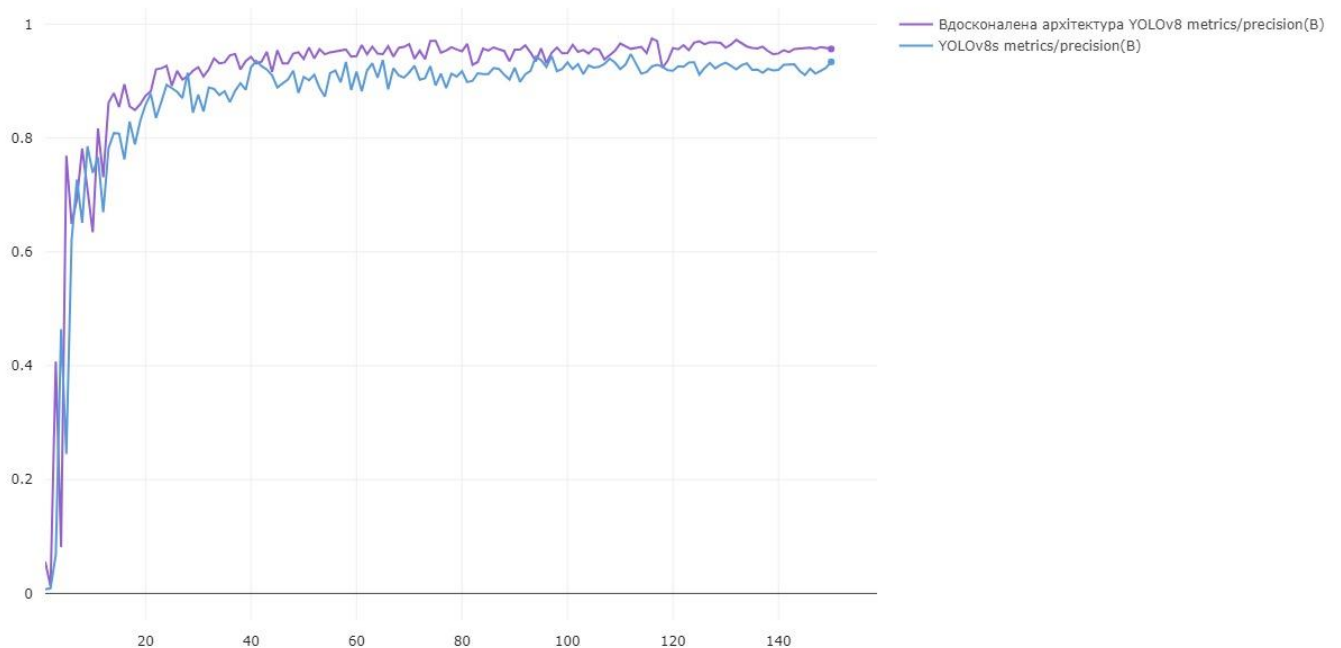


Рисунок 4.23 – Порівняння значень точності (P).

Точність вимірює долю правильно ідентифікованих об'єктів серед усіх об'єктів, які модель помітила як клас В.

З Рисунка 4.23 зрозуміло, що точність для обох моделей (фіолетової для вдосконаленої та синьої для стандартної) швидко зростає на ранніх етапах тренування, свідчачи про ефективність моделей у навчанні розпізнавати клас В. Після ініціального стрімкого зростання, обидві криві входять у більш плоску фазу, що вказує на стабілізацію точності.

Фіолетова крива знаходиться вище за синю на більшій частині графіка, що може свідчити про те, що вдосконалена архітектура YOLOv8 краще розпізнає клас В при меншій кількості помилкових позитивів у порівнянні зі стандартною моделлю.

Обидві моделі досягають точності вище 0.8, що є досить високим показником, особливо враховуючи складність завдання виявлення БПЛА. Стабілізація точності при високих значеннях свідчить про надійність моделей у виявленні класу В і може вказувати на те, що додаткове тренування не приносить суттєвого покращення.

Завдяки високій точності вдосконаленої моделі, система має потенціал бути використаною в практичних застосуваннях, де важливо мінімізувати кількість помилкових спрацьовувань при виявленні БПЛА.

Таблиця 4.3 – Результати експериментів на власному наборі даних.

<b>Model</b>	<b>P</b>	<b>R</b>	<b>mAP</b>
YOLOv8s	92%	77.2%	85.3%
Вдосконалена модель	95.8%	85.5%	91.8%

У порівнянні з результатами моделей на наборі даних TIB-Net, де найкращий mAP становив 92.1%, вдосконалена модель на власному наборі даних продемонструвала майже такий самий рівень точності.

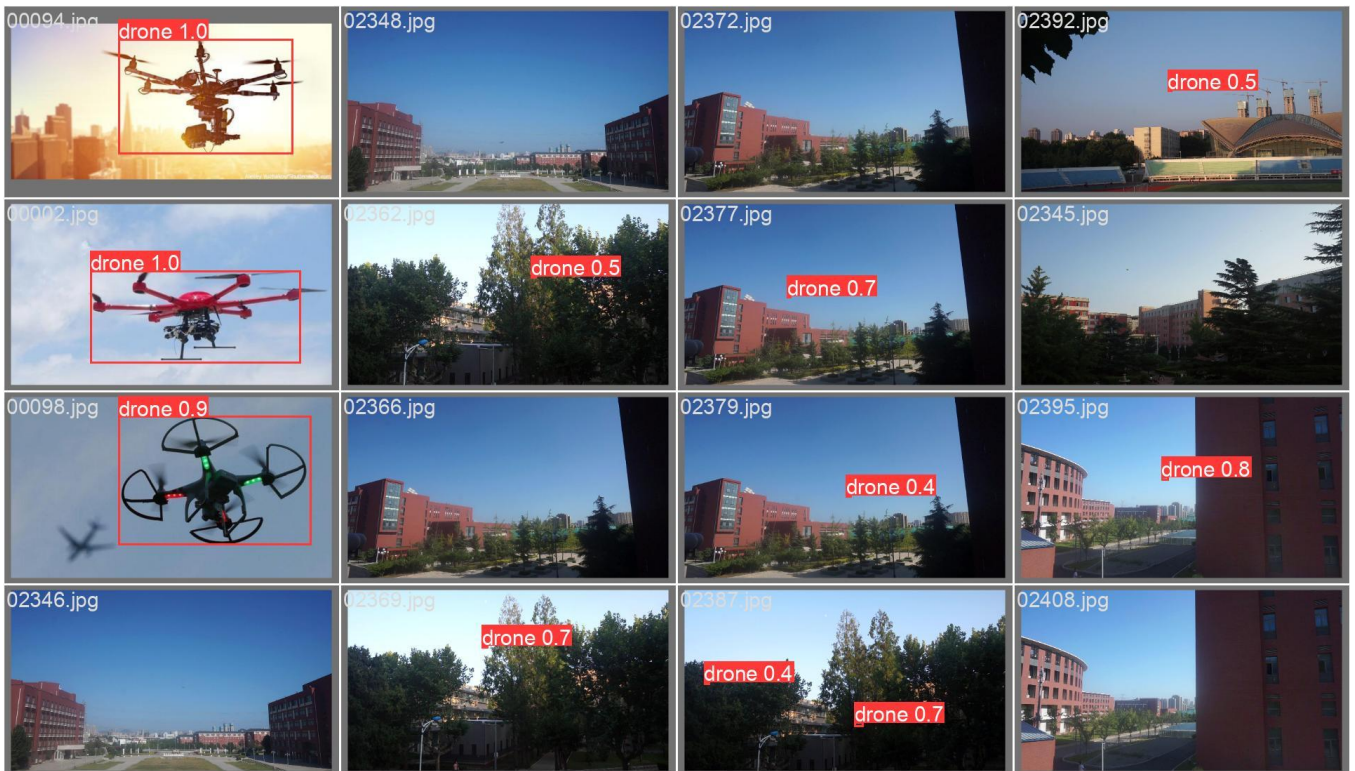


Рисунок 4.24 – Результат розпізнавання БПЛА моделлю YOLOv8s.

Проаналізуючи Таблицю 4.3 можна зробити висновок, що отриманий чудовий результат, враховуючи, що власний набір даних містив більше зображень і мав складніший задній фон, що зазвичай ускладнює завдання детекції.

Такі результати вказують на високу ефективність вдосконаленої моделі у виявленні БПЛА, незважаючи на складні умови власного набору даних. Це демонструє, що запропоновані вдосконалення в архітектурі мережі значно покращили її здатність до розпізнавання малих і візуально складних об'єктів у різноманітних умовах. Таким чином, отримані результати підтверджують високу ефективність запропонованого методу для застосування у практичних сферах, де необхідно швидко та точно виявлення БПЛА.

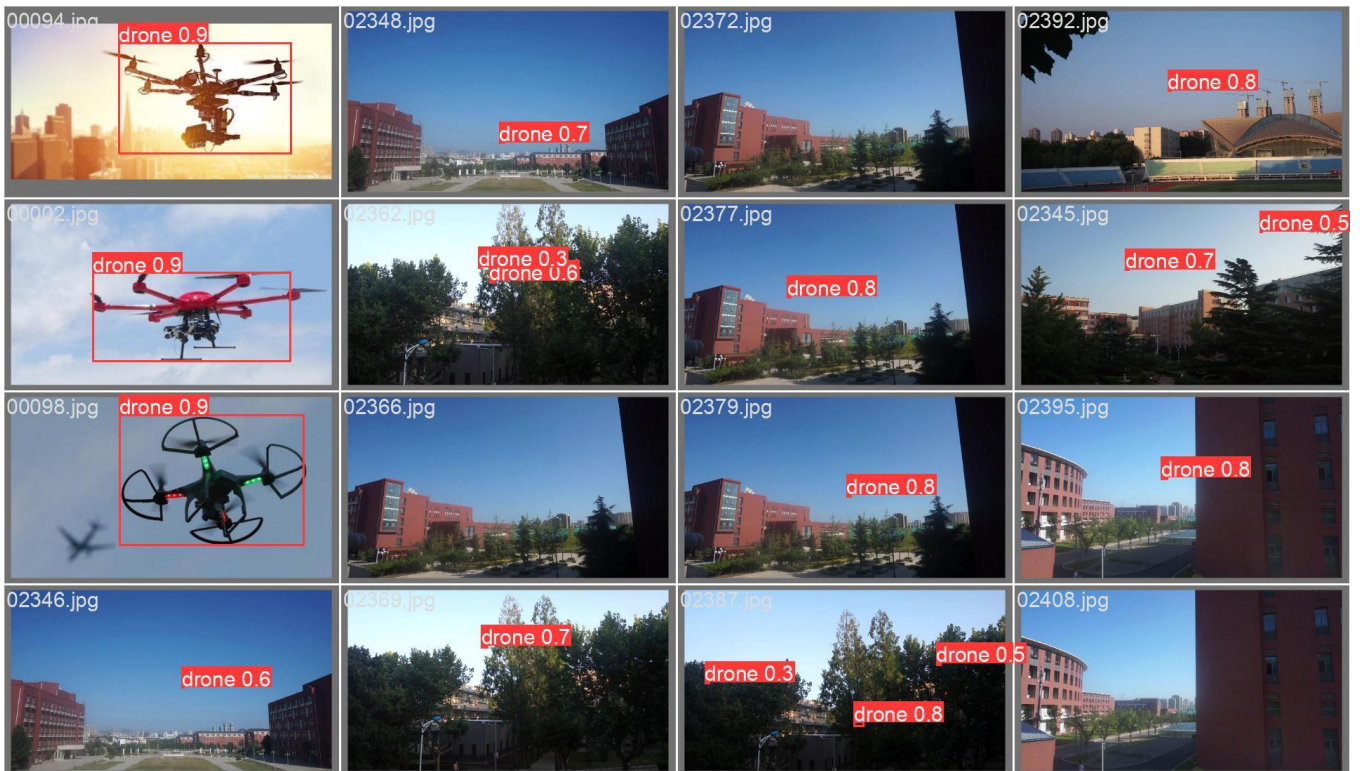


Рисунок 4.25 – Результат розпізнавання БПЛА вдосконаленою моделлю.

Порівнюючи Рисунок 4.24 та Рисунок 4.25 можна помітити, що результати вдосконаленої моделі є кращими в порівнянні із стандартною моделлю YOLOv8s. Але все ще присутні хибні виявлення. Це відбувається через складний фон зображення.

#### Висновки до розділу 4

В четвертому розділі виконано експериментальне дослідження вдосконаленого методу виявлення багатороторних БПЛА методом ШІ. Результати демонструють значне покращення в продуктивності моделі YOLOv8 завдяки впровадженню ряду модифікацій, зокрема додаванню детекційного шару P2 та використанню модулів Swin Transformer і GAM. Експерименти, проведені на загальнодоступному наборі даних UAV TIB-Net, підтверджують, що ці модифікації забезпечують значне покращення у виявленні мікроцілей БПЛА, що відображається у зростанні показників точності (P), повноти (R) та середньої точності (mAP).

Зокрема, додавання модулю Swin Transformer збільшує точність виявлення, а GAM – поліпшує якість злиття характеристик.

Незважаючи на підвищення обчислювальної складності, що веде до зменшення швидкості обробки кадрів (FPS), запропонована модель все ж забезпечує відповідність вимогам реального часу, що є критичним для практичного застосування. Крім того, впровадження цих нововведень дозволило значно скоротити кількість параметрів та розмір моделі, що сприяє більш ефективному використанню ресурсів та зменшенню вимог до обчислювальної потужності.

## ВИСНОВКИ

У цій роботі було запропоновано вдосконалений метод до виявлення багатороторних БПЛА з використанням методів штучного інтелекту.

Актуальність теми роботи обумовлена зростаючим використанням БПЛА у різних сферах, що підкреслює необхідність ефективних методів їх виявлення для забезпечення національної безпеки та запобігання потенційним загрозам.

Було проведено глибокий аналіз нейромережевих моделей та існуючих підходів для виявлення БПЛА, що дозволило ідентифікувати ключові аспекти для покращення. Сформульовано та реалізовано метод виявлення БПЛА з використанням модифікованої згорткової нейронної мережі YOLOv8, механізму GAM та Swin Transformer, забезпечуючи високу точність і ефективність.

Було проведено експериментальне дослідження вдосконаленого методу виявлення багатороторних БПЛА за допомогою ШІ. В результаті цього дослідження було досягнуто значного покращення в продуктивності моделі YOLOv8. Основні результати експериментів, представлені у Таблиці 4.2, включають наступні ключові показники:

Додавання детекційного шару P2 у модуль head покращило продуктивність моделі. Точність (P) покращилася на 6.5%, повнота (R) на 8.4% та середня точність (mAP) на 5.1%. Це свідчить про те, що додавання такого шару ефективно підвищує здатність моделі виявляти малі цілі.

Параметри моделі були скорочені, з 11.126 мільйонів у базовій моделі до 7.511 мільйонів у покращеній моделі. Це зниження кількості параметрів сприяє більш ефективному використанню ресурсів та зменшенню вимог до обчислювальної потужності.

Розмір моделі був зменшений з 21.9 МБ до 14.8 МБ, що також сприяє оптимізації використання пам'яті та обчислювальних ресурсів.

Швидкість обробки кадрів (FPS) зазнала змін, з 285 FPS у базовій моделі до 191 FPS у покращеній моделі, що все ще забезпечує відповідність вимогам реального часу, що є критичним для практичного застосування.

Ці результати демонструють, що запропоноване вдосконалення забезпечує значне покращення у виявленні мікроцілей БПЛА, що відображається у зростанні показників P, R та mAP. Отже, досягнуті результати підтверджують ефективність та практичність покращеної моделі для задач виявлення БПЛА.

### Перелік посилань

1. PROSPECTS OF THE DEVELOPMENT OF UNMANNED AERIAL VEHICLES (UAVs) [Електронний ресурс]. Режим доступу: [https://www.researchgate.net/publication/345306836\\_PROSPECTS\\_OF\\_THE\\_DEVELOPMENT\\_OF\\_UNMANNED\\_AERIAL\\_VEHICLES\\_UAVs](https://www.researchgate.net/publication/345306836_PROSPECTS_OF_THE_DEVELOPMENT_OF_UNMANNED_AERIAL_VEHICLES_UAVs).
2. Присяжнюк, О. І., Черняк, М. О., & Балагура, О. В. (2022). Теоретичні та практичні основи застосування БПЛА на селекційно-насінницьких посівах буряків цукрових. Новітні агротехнології, 10(3). Режим доступу: <https://doi.org/10.47414/na.10.3.2022.270502>
3. РОЗРОБКА КОНЦЕПЦІЇ ОПТИЧНОГО КАНАЛУ ЗАХИЩЕНОГО ЗВ'ЯЗКУ З КВАДРОКОПТЕРАМИ В УМОВАХ АКТИВНОЇ ПРОТИДІЇ [Електронний ресурс]. Режим доступу: <https://jarch.donnu.edu.ua/article/view/10573/10482>
4. Модель застосування безпілотних авіаційних комплексів в інтересах виконання вогневих завдань артилерією в збройних конфліктах [Електронний ресурс]. Режим доступу: <https://doi.org/10.33099/2311-7249/2022-45-3-83-90>
5. Different Types of Drones and Uses [Електронний ресурс]. Режим доступу: <https://www.jouav.com/blog/drone-types.html>
6. Colomina I. Unmanned aerial systems for photogrammetry and remote sensing: A review [Електронний ресурс]. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0924271614000501>.
7. «Війна дронів»: про ефективність застосування БПЛА на полі бою [Електронний ресурс]. – Режим доступу: <https://armyinform.com.ua/2022/08/04/vijna-droniv-pro-efektyvnist-zastosuvannya-bpla-na-poli-boyu/>
8. Рекомендації підрозділам щодо боротьби з безпілотними літальними апаратами “КАМІКАДЗЕ” “SHANED-136” (“ГЕРАНЬ-2”) [Електронний ресурс]. Режим доступу: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi0rdnQrteCAxWEhP0HHdRCB10QFnoECAoQAQ&url=https%3A%2F%2Fsprotvyg7>.

com.ua%2Flesson%2Frekomendacii-pidrozdilam-shhodo-borotbi-z-bezpilotnimi-litalnimi-aparatami-kamikadze-shahed-136-geran-2&usg=AOvVaw1h4glMvdImPZDBI5Mw7Cci&opi=89978449

9. Audio-Based Drone Detection and Identification Using Deep Learning Techniques with Dataset Enhancement through Generative Adversarial Networks [Электронный ресурс]. Режим доступа: <https://www.mdpi.com/1424-8220/21/15/4953>

10. Drone Detection and Tracking Using RF Identification Signals [Электронный ресурс]. Режим доступа: <https://www.mdpi.com/1424-8220/23/17/7650>

11. Real UAV-Bird Image Classification Using CNN with a Synthetic Dataset [Электронный ресурс]. Режим доступа: <https://doi.org/10.3390/app11093863>

12. Oleksandr Sotnikov, Vladimir Kartashov, Oleksandr Tymochko, Oleg Sergiyenko, Vera Tyrsa, Paolo Mercorelli, Wendy Flores-Fuentes. Methods for Ensuring the Accuracy of Radiometric and Optoelectronic Navigation Systems of Flying Robots in a Developed Infrastructure. Chapter 16// Machine Vision and Navigation; Editors: Sergiyenko, Oleg, Flores-Fuentes, Wendy, Mercorelli, Paolo; pp.537-578.

13. Kartashov, V., Oleynikov, V., Koryttsev, I., Zubkov, O., Babkin S., Sheiko, S. Processing and Recognition of Small Unmanned Vehicles Sound Signals. 2018 International Scientific-Practical Conference on Problems of Infocommunications. Science and Technology (PIC S and T 2018) – Proceedings, 31 January 2019; pp. 392-396.

14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

15. Jiao, L.C.; Zhang, F.; Liu, F.; Yang, S.Y.; Li, L.L.; Feng, Z.X.; Qu, R. A Survey of Deep Learning-Based Object Detection. IEEE Access 2019, 7, 128837–128868.

16. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone Detection Network with Tiny Iterative Backbone. IEEE Access 2020, 8, 130697–130707.

17. S. Hijazi. Convolutional Neural Networks for Image Recognition / S. Hijazi, R. Kumar [Электронный ресурс]. Режим доступа: [https://ip.cadence.com/uploads/901/cnn\\_wp-pdf](https://ip.cadence.com/uploads/901/cnn_wp-pdf)
18. He, J.; Liu, M.; Yu, C. UAV reaction detection based on multi-scale feature fusion. In Proceedings of the 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Xi'an, China, 28–30 October 2022; pp. 640–643.
19. Wastupranata, L.M.; Munir, R. UAV Detection using Web Application Approach based on SSD Pre-Trained Model. In Proceedings of the 2021 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Virtual, 3–4 November 2021; pp. 1–6.
20. Tao, Y.; Zongyang, Z.; Jun, Z.; Xinghua, C.; Fuqiang, Z. Low-altitude small-sized object detection using lightweight feature-enhanced convolutional neural network. *J. Syst. Eng. Electron.* 2021, 32, 841–853.
21. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An Efficient Network for Low-Altitude Object Detection Based on Convolution and Transformer. *IEEE Trans. Instrum. Meas.* 2022, 71, 2507412.
22. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* 2023, 111, 257–276.
23. Ma, J.; Yao, Z.; Xu, C.; Chen, S. Multi-UAV real-time tracking algorithm based on improved PP-YOLO and Deep-SORT. *J. Comput. Appl.* 2022, 42, 2885.
24. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
25. Sunkara, R.; Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. *arXiv* 2022, arXiv:2208.03641.
26. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* 2021, arXiv:2112.05561.
27. Indoor objects detection and recognition for an ICT mobility assistance of visually impaired people [Электронный ресурс]. Режим доступа:

[https://www.researchgate.net/publication/343813921\\_Indoor\\_objects\\_detection\\_and\\_recognition\\_for\\_an\\_ICT\\_mobility\\_assistance\\_of\\_visually\\_impaired\\_people](https://www.researchgate.net/publication/343813921_Indoor_objects_detection_and_recognition_for_an_ICT_mobility_assistance_of_visually_impaired_people)

28. Hossain D., Imtiaz M. H., Ghosh T., Bhaskar V., Sazonov, E. Real-Time Food Intake Monitoring Using Wearable Egocentric Camera. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2020, P. 4191–4195. DOI: 10.1109/EMBC44109.2020.9175497.
29. Thadeshwar H., Shah V., Jain M., Chaudhari R., Badgujar V. Artificial Intelligence based Self-Driving Car. 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP). 2020. P. 1–5. DOI: 10.1109/ICCCSP49186.2020.9315223.
30. Howard G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv. 17 Apr 2017. P. 1–9. DOI: 10.48550/arXiv.1704.04861.
31. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. P. 4510–4520. DOI: 10.1109/CVPR.2018.00474.
32. Understanding Attention Modules: CBAM and BAM — A Quick Read [Электронный ресурс]. Режим доступа: <https://medium.com/visionwizard/understanding-attention-modules-cbam-and-bam-a-quick-read-ca8678d1c671>
33. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, An image is worth 16x16 words: Transformers for image 95 recognition at scale, 2020. DOI: 10.48550/ARXIV.2010.11929 [Электронный ресурс] – Режим доступа до ресурсу: <https://arxiv.org/abs/2010.11929>.
34. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, 2017. DOI: 10.48550/ARXIV.1706.03762 [Электронный ресурс] – Режим доступа до ресурсу: <https://arxiv.org/abs/1706.03762>.

35. Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan and Z. He, A survey of visual transformers, 2021. arXiv: 2111.06091 [cs.CV].
36. D. Hendrycks and K. Gimpel, Gaussian error linear units (gelus), 2016. DOI: 10.48550/ARXIV.1606.08415 [Электронный ресурс] – Режим доступа до ресурсу: <https://arxiv.org/abs/1606.08415>.
37. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, 2021. DOI: 10.48550/ARXIV.2103.14030 [Электронный ресурс] – Режим доступа: <https://arxiv.org/abs/2103.14030.96>
38. Loy, J., A Comprehensive Guide to Microsoft's Swin Transformer. In-depth Explanation and Animations, 20.05.2022 [Электронный ресурс] – Режим доступа: <https://towardsdatascience.com/a-comprehensive-guide-to-swin-transformer64965f89d14c>.
39. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. [Электронный ресурс] – Режим доступа: [https://scholar.google.com/scholar\\_lookup?title=Global+attention+mechanism:+Retain+information+to+enhance+channel-spatial+interactions&author=Liu,+Y.&author=Shao,+Z.&author=Hoffmann,+N.&publication\\_year=2021&journal=arXiv](https://scholar.google.com/scholar_lookup?title=Global+attention+mechanism:+Retain+information+to+enhance+channel-spatial+interactions&author=Liu,+Y.&author=Shao,+Z.&author=Hoffmann,+N.&publication_year=2021&journal=arXiv)
40. Liu, S.; Wang, Y.; Yu, Q.; Liu, H.; Peng, Z. CEAM-YOLOv7: Improved YOLOv7 Based on Channel Expansion and Attention Mechanism for Driver Distraction Behavior Detection. IEEE Access 2022, 10, 129116–129124. [Электронный ресурс] – Режим доступа: <https://doi.org/10.1109/ACCESS.2022.322833>
41. Integrated Learning-Based Pest and Disease Detection Method for Tea Leaves [Электронный ресурс] – Режим доступа: <https://www.mdpi.com/1999-4907/14/5/1012>

# ДОДАТКИ

**Додаток А**  
**Довідка ВХНУ ТН 17/11/23**

**Довідка:** ВХНУ ТН 17/11/23

**Видання:** Вісник Хмельницького національного університету. Технічні науки

**Категорія фаховості видання:** фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрямок – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

**Назва статті:** МЕТОД ВИЯВЛЕННЯ БАГАТОРОТОРНИХ БПЛА ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ

**Автори:** Сергій Горелов, Олександр Бармак, Едуард Манзюк (Хмельницький національний університет»)

**Номер, у який прийнято статтю:** №6 до друку рекомендовано буде до 25 грудня 2023 року.

17.11.2023

Начальник відділу  
інтелектуальної власності та трансферу технологій

Ю.В.Кравчик



І.С.Мартинюк

## **Додаток Б**

### **Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра**

*(ксерокопії титульної сторінки, сторінки змісту та всіх сторінок із публікацією)*

#### **Перелік наукових публікацій:**

Горелов С.В., Бармак О.В., Манзюк Е.А. Метод виявлення багатороторних БПЛА засобами штучного інтелекту. Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023. №6

УДК 004.8

DOI:

**ГОРЄЛОВ С.В., БАРМАК О.В., МАНЗЮК Е.А.**

Хмельницький національний університет

e-mail: shorielov@ukr.net

**МЕТОД ВИЯВЛЕННЯ БАГАТОРОТОРНИХ БПЛА ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ**

*Зростаюче використання безпілотних літальних апаратів (БПЛА) у різноманітних комерційних та технічних секторах підкреслило необхідність надійного спостереження за БПЛА, особливо з огляду на питання громадської безпеки. У відповідь на цю вимогу було швидко розвинуто методи ідентифікації БПЛА. Незважаючи на прогрес, маленький розмір дронів, складні умови повітряного простору та зміни у освітленні залишаються значними викликами в цій сфері досліджень. З метою розв'язання цих викликів проведено дослідження, що базується на новому підході до виявлення БПЛА невеликих розмірів, використовуючи вдосконалену версію YOLOv8.*

*Запропоновано метод, який дозволяє інтегрувати модуль високої деталізації для підвищення точності виявлення дрібних повітряних об'єктів. Також було оптимізовано архітектуру шляхом видалення елементів, що є більш важливими для великих об'єктів, тим самим зменшуючи обчислювальне навантаження та прискорюючи швидкість виявлення БПЛА. Крім того, використання Swin Transformer дозволяє краще виявляти об'єкти різних масштабів та зменшує обчислювальну складність. Також впровадження модуля уваги GAM у архітектуру мережі значно підвищує здатність до асиміляції ознак, що призводить до помітного підвищення ефективності виявлення БПЛА.*

*У порівнянні з оригінальною базовою моделлю, наш запропонований метод демонструє суттєві покращення, збільшуючи P (precision), R (recall) та mAP (mean average precision) на 9%, 10,2% та 6% відповідно. Крім того, він істотно оптимізує модель за кількістю вхідних параметрів на 32,5% та 32,4% відповідно. Запропонований метод у порівняльних експериментах та експериментах зі специфічними наборами даних показав велику перспективу для практичної реалізації в системах виявлення об'єктів БПЛА, що дає підстави говорити про перспективу застосування в реальних умовах.*

*Ключові слова: БПЛА, виявлення об'єктів, YOLOv8, глибоке навчання.*

**Serhii HORIELOV, Olexander BARMAK, Eduard MANZIUK**

Khmelnytskyi National University

**METHOD OF DETECTING MULTI-ROTOR UAVS USING ARTIFICIAL INTELLIGENCE**

*The escalating deployment of Unmanned Aerial Vehicles (UAVs) in various commercial and technical sectors has highlighted the need for robust UAV surveillance, particularly for public safety considerations. In response to this demand, there has been rapid advancement in methods for identifying UAVs. Despite the progress, the diminutive nature of drones, the intricate environments of aerial space, and fluctuating lighting conditions continue to present considerable challenges in this field of study. Addressing these issues, our study introduces an innovative detection approach for small UAVs, utilizing an enhanced version of YOLOv8.*

*Our strategy involves integrating a high-definition detection module to augment the accuracy of spotting diminutive aerial objects. Concurrently, we streamline the architecture by removing components less critical for larger objects, thereby diminishing the computational load and boosting UAV detection speed. Additionally, the use of the Swin Transformer enables better detection of objects of varying scales and reduces computational complexity. The*

*implementation of the GAM attention module into the network architecture significantly enhances feature assimilation, leading to a noticeable increase in UAV detection efficiency.*

*Compared to the original baseline model, our proposed method shows substantial improvements, increasing P (precision), R (recall), and mAP (mean average precision) by 9%, 10.2%, and 6% respectively. Furthermore, it significantly optimizes the model and its size by 32,5% and 32,4%. The proposed method, in comparative experiments and experiments with specific datasets, has shown great potential for practical implementation in UAV detection systems, suggesting its feasibility for real-world application.*

*Keywords: UAV; object detection; YOLOv8; deep learning.*

### **Постановка проблеми**

Тема розпізнавання багатороторних безпілотних літальних апаратів (БПЛА) засобами штучного інтелекту в останні роки набула великої актуальності у сучасному науково-технічному середовищі. Зростання популярності БПЛА спостерігається у різноманітних сферах життєдіяльності – від аграрної та геодезійної галузей до використання в мілітарних цілях і системах нагляду. Це зумовлено різноманітністю їх застосувань, здатністю виконувати завдання в недоступних або небезпечних для людей зонах, а також їх ефективністю та доступністю.

У зв'язку з цим, ідентифікація та моніторинг БПЛА стають критично важливими задачами, особливо у контексті забезпечення національної безпеки та приватності. Розпізнавання багатороторних БПЛА методами штучного інтелекту відіграє ключову роль у вирішенні цього питання. Штучний інтелект, а конкретно – машинне навчання та нейронні мережі, демонструють значні результати в детектуванні та класифікації об'єктів у реальному часі, навіть у складних умовах.

Актуальність дослідження посилюється високою здатністю сучасних методів штучного інтелекту адаптуватися до динамічних умов реального середовища, вирішувати завдання з великою точністю та ефективністю, що є умовою для покращення систем розпізнавання БПЛА.

### **Аналіз останніх джерел**

Зростання продуктивності апаратних засобів підвищило здатність обробки даних комп'ютерами, що призвело до стрімкого прогресу в технологіях обробки візуальних даних, які базуються на глибокому навчанні з використанням великого набору даних. Виявлення об'єктів за допомогою технологій комп'ютерного зору привернуло значну увагу дослідників. Технологія розвивалася від традиційного виявлення ознак [1, 2, 3], що використовували обчислення згорток для розпізнавання об'єктів, до використання глибокого навчання для підвищення точності розпізнавання при візуальному представленні об'єктів. Порівняно з традиційними методами виявлення електромагнітних сигналів, такими як радар, лазер, інфрачервоне випромінювання, аудіо та радіочастоти, виявлення об'єктів за допомогою візуальних сенсорів, зокрема камер, що записують групові відео та зображення, пропонує більш інтуїтивне виявлення. Виявлення об'єктів за допомогою візуальних сенсорів, має переваги, такі як виявлення у реальному часі та динамічний запис послідовних зображень цілей, низька вартість, висока швидкість виявлення та стійкість до низьковисотних перешкод [4].

Виявлення об'єктів є важливою сферою досліджень в області комп'ютерного зору та основою для чисельних, складних, візуальних завдань. Його широко застосовують у промисловості, сільському господарстві та інших галузях [5,6]. З 2014 року було досягнуто значного прогресу в методах виявлення об'єктів, заснованих на глибокому навчанні. В промисловості було введено різноманітні алгоритми, включаючи Faster R-CNN [7], SSD [8] і серію YOLO [9]. Зі швидким розвитком технології виявлення цілей з'явилося кілька ефективних методів, які можна застосувати для завдань виявлення БПЛА [10,11,12,13,14]. Наприклад, у дослідженні [10] яке присвячене детектуванню обличчя на зображенні, вказано, що згорткові нейронні мережі стикаються з викликом забезпечення балансу між точністю виявлення та розміром моделі. Для вирішення цієї проблеми автори запропонували рекурентний шлях і модуль просторової уваги в оригінальний детектор обличчя (EXTD), покращуючи його здатність вилучати характеристики з маленьких цілей БПЛА. Розмір моделі складає лише 690,7 кБ. Однак, ця модель показує повільний час виведення та не підходить для використання у практичних

інженерних сценаріях. У дослідженні [11] було запропоновано мережу виявлення цілей БПЛА на основі інтегрального об'єднання функцій: спочатку вилучаються характеристики чутливого поля цілі за допомогою res2net, далі покращується продуктивність мережі вилученням деяких та злиттям інших характеристик, і нарешті досягається кращий результат на самостійно побудованому наборі даних для виявлення БПЛА. У роботі [12] авторами було наведено новий метод виявлення БПЛА, цей метод покращує процес ідентифікації БПЛА, зосереджуючись на параметрах та обчислювальних можливостях для забезпечення виявлення в режимі реального часу через веб-додаток. Це дозволяє отримувати дані про присутність БПЛА у визначеній зоні з будь-якого місця, навіть через мобільний телефон. Експерименти показують, що використання веб-додатку надає переваги у порівнянні з традиційними методами обробки, і процес виявлення є значно ефективнішим. У роботі [13] пропонується легка згортова нейронна мережа з розширеними функціями, здатна до високоточного виявлення об'єктів низького польоту у реальному часі. Вона ефективна проти нелегальних дронів у повітряному просторі. У роботі [14] пропонується новий метод глибокого навчання, який називається мережею згорткових перетворень – Convolutional and Transformation Network (CT-Net). Основа цієї мережі спочатку включає блок трансформації що підсилює увагу, він створює підсилений механізм для покращення здатності моделі вилучати характеристики. Потім застосовується легкий модуль bottleneck для контролю обчислювального навантаження та зниження обчислювальних вимог. Також, пропонується структура злиття інформації з різних джерел або каналів для покращення точності виявлення при роботі з об'єктами різного масштабу, особливо малими об'єктами. Підхід досягає mAP 0,966 на самостійно створеному наборі даних для маленьких об'єктів низької висоти, демонструючи високу точність виявлення. Проте, встановлений показник швидкості обробки зображень у системі, складає лише 37 кадрів на секунду. Для виявлення БПЛА, швидкість обробки зображень є критичним фактором. Оскільки БПЛА часто рухаються з високою швидкістю і можуть з'явитися в різноманітних середовищах, системам виявлення необхідно швидко аналізувати візуальну інформацію, щоб ефективно ідентифікувати і відстежувати ціль. Швидкість в 37 кадрів на секунду (frames per second - FPS) може бути недостатньою для деяких сценаріїв використання, де важлива миттєва реакція. Тому, існуючий показник FPS вказує на потенціал для розвитку та оптимізації системи виявлення БПЛА, з метою досягнення більш високої швидкості обробки кадрів без втрати точності виявлення.

Хоча у технології виявлення БПЛА було досягнуто значного прогресу, існуючі методи виявлення все ще стикаються з проблемами балансування точності виявлення, розміру моделі та швидкості виявлення. Мережа виявлення серії YOLO ефективно вирішила ці проблеми. Моделі серії YOLO пройшли вісім офіційних ітерацій та кілька версій, продемонструвавши чудову точність та швидкодію виявлення. Ці моделі мають широке застосування у різних галузях, включаючи медицину, транспорт, дистанційне зондування та промисловість [15]. Вчені широко досліджували використання моделей серії YOLO для виявлення цілей БПЛА, як це відображено у численних працях [16,17,18,19]. Наприклад, у дослідженні [16], шляхом інтеграції модуля механізму уваги в алгоритм виявлення PP-YOLO, були зроблені вдосконалення для покращення його продуктивності. Крім того, введення функції активації Mish, вирішило проблему градієнт-зникнення під час процесу зворотного розповсюдження, що призвело до значного підвищення точності виявлення. У роботі [17] було запропоновано алгоритм виявлення БПЛА для складних міських фонів, на основі YOLOv3. Він використовував популярну архітектуру Feature Pyramid Network (FPN), яка ефективно використовує інформацію на різних рівнях роздільної здатності, щоб покращити здатність системи виявляти об'єкти незалежно від їхнього розміру в зображенні. Також була використана мережа Ghost для прискорення моделі. Експериментальні результати показали, що алгоритм виявив високу стійкість та ефективно виявляє малі цілі БПЛА у складних сценах. У дослідженні [18] згортова нейронна мережа, MobileNetv2, замінила оригінальний backbone CSPDarknet53, високопродуктивної моделі YOLOv4. Ця заміна була спрямована на зменшення масштабу моделі та спрощення обчислювальних операцій. Експериментальні результати показали, що Mob-YOLO може досягти точного моніторингу цілей БПЛА в реальному часі з меншим розміром моделі, що робить її придатною для вбудованих процесорів. У роботі [19] було запропоновано розподілену антидронову систему на основі YOLOv5. Об'єднуючи ідеї поліпшення, запропоновані у вищевказаній літературі щодо серії YOLO,

авторами пропонується вдосконалити модель YOLOv8s та зробити її придатною для виявлення маленьких об'єктів БПЛА, без суттєвих втрат точності виявлення та швидкості на складному наборі даних про малі БПЛА.

**Мета роботи** полягає в удосконаленні моделі YOLOv8s для покращення виявлення безпілотних літальних апаратів (БПЛА) шляхом адаптації архітектури мережі до конкретних особливостей практичного застосування, зокрема виявлення об'єктів малих розмірів.

#### Виклад основного матеріалу

Зважаючи на проведений аналіз за основу було обрано YOLOv8. YOLOv8 є еволюцією попередніх версій YOLO, включаючи нові характеристики та удосконалення для подальшого підвищення продуктивності та гнучкості, досягаючи вражаючої швидкості та високої ефективності. YOLOv8 пропонує п'ять моделей різного розміру: нано, мала, середня, велика та дуже велика. Наномодель має лише 3,2 мільйони параметрів, що сприяє легкому впровадженню на мобільні пристрої та пристрої, які працюють лише на CPU. Щоб знайти баланс між точністю виявлення та швидкістю, в цій роботі використовується модель YOLOv8s для виявлення БПЛА, яка є результатом розширення та поглиблення структури наномережі. YOLOv8 поділяється на backbone, neck та head, які використовуються для екстракції особливостей, злиття багатьох особливостей та виведення прогнозу. Дизайн мережі YOLOv8 представлений на Рисунок 1.

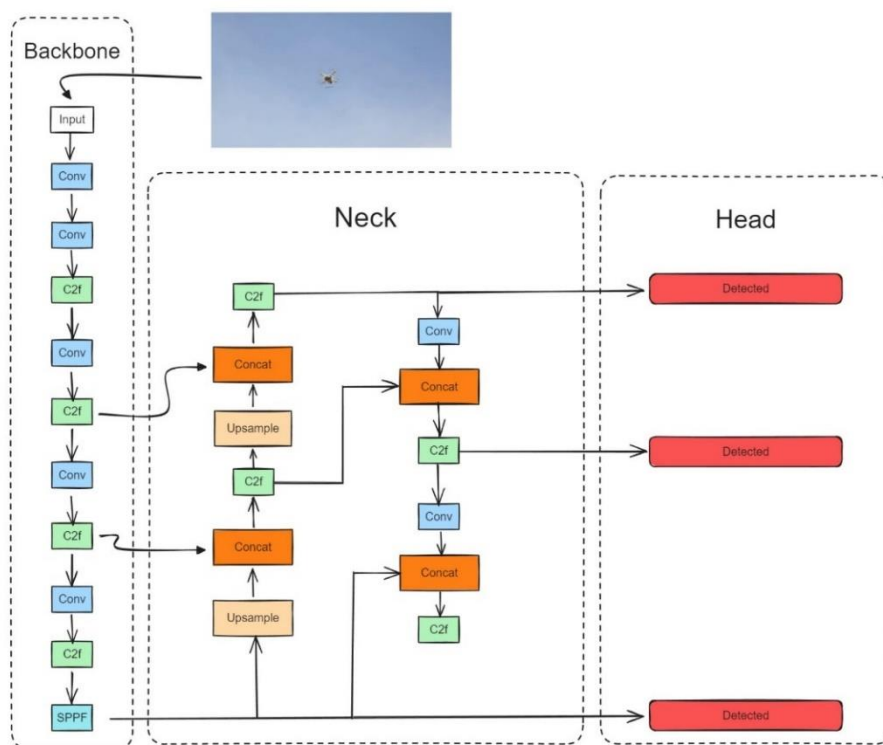


Рисунок 1 – Структура мережі YOLOv8.

Мережа екстракції особливостей в основному вилучає особливості окремих масштабів з зображень за допомогою модулів C2f та SPPF. Модуль C2f, в порівнянні з C3 модулем, який використовувався у попередніх версіях мережі, зменшує кількість шарів мережі на один згортковий шар, роблячи модель легшою. Також він

інтегрує переваги структури ELAN з YOLOv7, ефективно розширюючи гілку градієнтів за допомогою bottleneck модулів для отримання більш багатой інформації про потік градієнтів [20]. SPPF зменшує кількість шарів мережі на основі SPP (spatial pyramid pooling - "піраміда" з узагальнюючих шарів) [21], щоб усунути зайві операції та виконувати швидше злиття особливостей. Модуль багаторівневого злиття використовує комбінацію FPN (feature pyramid network - згортова нейронна мережа, побудована у вигляді піраміди) [22] та PAN (path aggregation network - мережа агрегації шляхів) [23]. Шляхом двостороннього злиття особливостей низького та високого рівня він підсилює особливості низького рівня з меншими полями сприйняття та покращує здатність виявлення цілей різних масштабів. Рівень виявлення передбачає позиції цілей, категорії, оцінки впевненості та іншу інформацію. YOLOv8 працює без використання якорів (Рисунок 3). Це означає, що прогнозується центр об'єкта, а не зміщення від відомого якорного блоку. У попередніх версіях YOLO якорні блоки були відомою складною частиною, оскільки вони могли представляти розподіл блоків цільового бенчмарку, але не розподіл призначеного для користувача набору даних.

Безякорне виявлення зменшує кількість прогнозів блоків, що пришвидшує процес неадекватного пригнічення (NMS), складний етап постобробки, який відбирає кандидатські детекції після виведення. В кінцевому підсумку, вона виконує багатомасштабні прогнози, використовуючи характеристики, зменшені у  $8\times$ ,  $16\times$ , та  $32\times$ , для досягнення точних прогнозів для малих, середніх і великих цілей. Детальні модулі в мережі YOLOv8 ілюстровані на Рисунок 2.

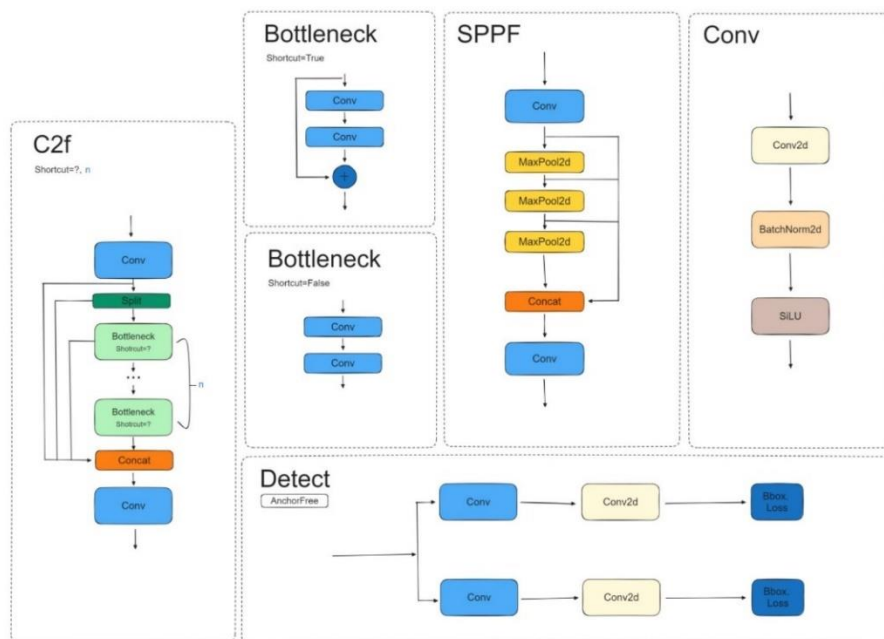


Рисунок 2 – Структурна схема мережі YOLOv8.

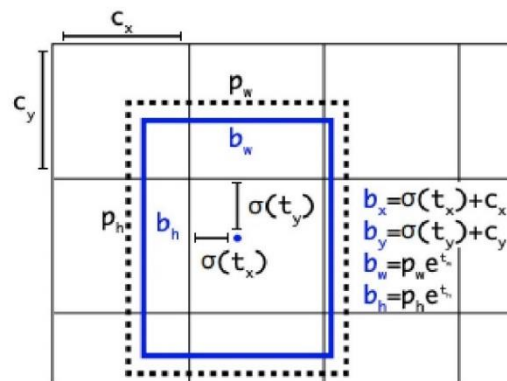


Рисунок 3 – Візуалізація якірного блоку в YOLO

YOLOv8 вилучає характеристики об'єктів, використовуючи глибоку залишкову мережу. Вона завершує багатомасштабний прогноз використовуючи структуру PAN, але YOLOv8 все ще виконує три ітерації зменшення при екстракції характеристик, щоб отримати максимальну карту характеристик. Однак, значна частина інформації про характеристики цілі втрачається, що могло б бути корисним для виявлення малих цілей. Таким чином, ця стаття вдосконалює YOLOv8 і пропонує модель мережі для виявлення маломасштабних цілей, а удосконалена структура мережі зображена на Рисунок 4. Удосконалення полягає у наступних кроках:

1. Підвищення здатності моделі до виявлення крихітних цілей, за допомогою інтеграції шару високої роздільної здатності в частину head та вилучення шару виявлення та пов'язаних з ним шарів екстракції та комбінування характеристик для прогнозування великих цілей.
2. Модуль backbone було вдосконалено шляхом інтеграції Swin Transformer(C3STR) [24].
3. Механізм уваги GAM [25] було введено у модуль комбінування багаторівневих характеристик(Concat), це необхідно щоб посилити злиття цільових характеристик моделі.

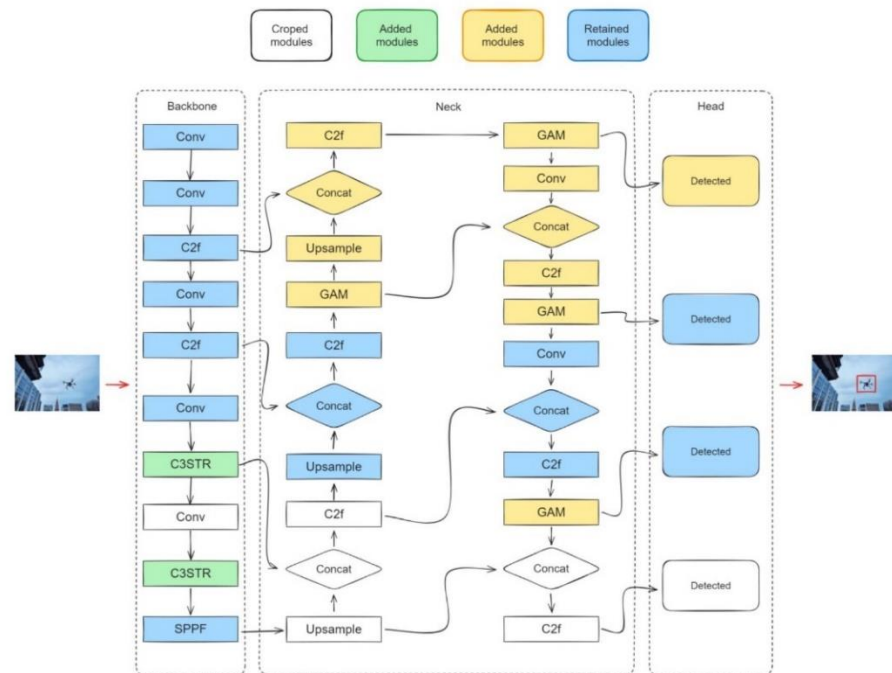


Рисунок 3 – Діаграма удосконаленої структури мережі YOLOv8.

#### Аналіз ефективності запропонованого методу

Для оцінки продуктивності моделі було використано загальнодоступний набір даних UAV TIB-Net [10]. Набір даних UAV TIB-Net складається з 2850 зображень, які демонструють різні типи БПЛА, включаючи багатороторні БПЛА та БПЛА з фіксованим крилом. Зображення були зняті фіксованою камерою на землі на відстані близько 500 м від аеродромів, а роздільна здатність зібраних зображень становила  $1920 \times 1080$  пікселів. Ці сцени охоплюють кілька низьковисотних сцен (небо, дерева, будівлі тощо) з зображень польотів БПЛА, повністю враховуючи зразки у різний час дня та в різних погодних умовах.

Апаратна конфігурація, використана для експериментів - це графічна карта NVIDIA GeForce RTX 3060 з 12 ГБ, фреймворк глибокого навчання PyTorch версії 1.13.1, Python версії 3.7.15, CUDA версії 11.7 та операційна система Ubuntu 22.04.

Функції втрат удосконаленої версії YOLOv8 відповідають YOLOv8 і включають втрату прямокутної коробки ( $Loss_{box}$ ), фокусовану втрату розподілу ( $Loss_{dfl}$ ) та втрату класифікації ( $Loss_{cls}$ ).

$$Loss = a \cdot Loss_{box} + b \cdot Loss_{dfl} + c \cdot Loss_{cls} \quad (1)$$

Параметри які є ваговими коефіцієнтами ( $a$ ,  $b$  і  $c$ ) представляють відповідну пропорцію ваги функції втрат у загальній функції втрат. У цьому експерименті вони становлять  $a = 7.5$ ,  $b = 1.5$  та  $c = 0.5$  відповідно.

Перед тренуванням зображення та мітки набору даних розподіляються на навчальний набір, набір для перевірки та набір для тестування у співвідношенні 7:1:2. Максимальна кількість епох для тренування набору даних встановлена в 150, з першими трьома епохами, використаними для тренування розігріву. Для регулювання швидкості навчання використовується стратегія оптимізації SGD з початковою швидкістю навчання 0.01. Враховуючи наявність численних дрібних об'єктів на зразкових зображеннях та необхідність балансу між реальним часом та точністю у процесі детектування, розмір зразка нормалізується до  $640 \times 640$ .

Цей розмір дозволяє розгортати модель без надто великих втрат корисної інформації з зображень. Для того щоб мати можливість порівняти продуктивності моделей у експериментах ми не будемо використовувати навчені ваги. Крім того, всі процеси тренування мають однакові налаштування. Основні параметри для процесу тренування показані у Таблиці 1.

Таблиця 1 – Таблиця параметрів.

Параметри	Значення
Epochs	150
Warmup-epochs	3
Warmup-momentum	0.8
Batch Size	8
Imgsize	640
Initial Learning Rate	0.01
Final Learning Rate	0.01
Patience	50
Optimizer	SGD
NMSIoU	0.7
Momentum	0.937
Mask-ratio	4
Weight-Decay	0.0005

Щоб перевірити продуктивність моделі, в якості експериментальних показників оцінювання вибрано  $P$ ,  $R$ ,  $AP$ ,  $mAP$ , кількість параметрів, розмір моделі та кадри в секунду (FPS).

Точність ( $P$  - assigasy) та повнота ( $R$  - recall) розраховуються таким чином:

$$P = \frac{TP}{(TP + FP)} \cdot 100\% \quad (2)$$

$$R = \frac{TP}{(TP + FN)} \cdot 100\% \quad (3)$$

де  $TP$  (true positives) позначає кількість правильно виявлених цілей,  $FP$  (false positives) позначає кількість фонів, виявлених як цілі, і  $FN$  (false negatives) позначає кількість цілей, виявлених як фон.

Середня точність ( $AP$ ) та середня точність середніх значень ( $mAP$ ) розраховуються таким чином:

$$AP = \int_0^1 p(r) d(r) \quad (4)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

де  $N$  — кількість категорій, а  $AP$  — середня точність кожної категорії. У нашому випадку ми маємо лише одну категорію - БПЛА, тому  $N = 1$ .

На основі набору даних UAV TIB-Net, були проведені експерименти з виявлення БПЛА, для дослідження впливу кожного доданого або модифікованого модуля на загальну продуктивність моделі. Починаючи з оригінального YOLOv8s як базової лінії, покращення head, backbone та neck було розподілено по

порядку. Для аналізу покращення продуктивності кожного модуля були визначені базова Модель 1, покращена Модель 2 (з доданою tiny-head), покращена Модель 3 (додана tiny-head і обрізана large-head), покращена Модель 4 (з доданою tiny-head, обрізаною large-head і доданим C3SPR), покращена Модель 5 (з доданою tiny-head, обрізаною large-head та доданим GAM), та покращена Модель 6 (з доданою tiny-head, обрізаною large-head, C3SPR та GAM). Зміни показників оцінювання для цих шести моделей були кількісно досліджені, і оптимальні результати для кожного показника оцінювання були виділені. Експериментальні результати моделей на наборі даних TIB-Net показані в Таблиці 2.

Таблиця 2 – Результати експериментів з вилученням.

Компоненти	1	2	3	4	5	6
+Tiny-Head		+	+	+	+	+
-Large-Head			+	+	+	+
+C3SPR				+		+
+GAM					+	+
P	81.4%	87.9%	89.1%	89.5%	88.2%	90.4%
R	78.1%	86.5%	86.5%	88.1%	86.2%	88.3%
mAP	86.1%	91.2%	91.5%	91.7%	90.6%	92.1%
Parameters/million	11.126	10.852	3.527	6.83	8.09	7.511
Model Size/MB	21.9	22.1	13.1	13.9	13.4	14.8
FPS/f.s-1	285	196	226	199	213	191

### Висновки

Для вирішення проблеми, виявлення БПЛА малих розмірів, запропоновано вдосконалену модель виявлення YOLOv8. Модель успішно адаптується до складнощів, пов'язаних з розміром дронів, характеристиками фону повітряного простору, рівнем світлової інтенсивності та іншими зовнішніми чинниками, що впливають на процес виявлення. Для покращення здатності виявляти дрібні об'єкти, спочатку, у частині head, було додано шар високої роздільної здатності для покращення здатності виявляти невеликі об'єкти. У той же час, large-head та надлишкові шари мережі було обрізано для ефективного зменшення кількості параметрів мережі та підвищення швидкості виявлення дронів. Нарешті, механізм уваги GAM було введено у писк для покращення злиття ознак цілей моделі, таким чином покращуючи загальну продуктивність моделі для виявлення дронів. Було проведено порівняльні експерименти на складному наборі даних TIB-Net. У порівнянні з початковою базовою моделлю YOLOv8, запропонований нами метод покращує продуктивність на 9%, 10,2% та 6% за показниками P (precision), R (recall) та mAP (mean average precision) відповідно, а також зменшує кількість параметрів моделі на 32,5% та зменшення розміру моделі на 32,4%. Крім того, модель виявлення показала кращі результати у порівняльних експериментах і експериментах з самоствореним набором даних. Отже, запропонований метод більше підходить для інженерного впровадження та практичного застосування систем виявлення цілей БПЛА.

Однак, додавання додаткових рівнів обробки у модуль head та використання як Swin Transformer, так і GAM модулів, призвело до збільшення часу висновку моделі та FPS знизився порівняно з базовою моделлю. Крім того, з експериментів на самоствореному наборі даних видно, що R знижується, коли фон повітряного простору є більш складним, тобто ймовірність пропуску виявлення зростає. Наступні роботи будуть присвячені покращенню точності виявлення в більш складних фонових умовах повітряного простору, одночасно зменшуючи час прийняття рішення моделлю.

### Літэратура

1. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 2004, 60, 91–110.
3. Bay, H.; Tuytelaars, T.; van Gool, L. Surf: Speeded up robust features. In Proceedings of the Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part I 9. Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
4. Dai, J.; Wu, L.; Wang, P. Overview of UAV Target Detection Algorithms Based on Deep Learning. In Proceedings of the 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 17–19 December 2021; Volume 2, pp. 736–745.
5. Zuo, Y. Target Detection System of Agricultural Economic Output Efficiency Based on Kruskal Algorithm. In Proceedings of the 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNBC), Tumkur, India, 2–3 December 2022; pp. 1–5.
6. Li, S.; Yu, J.; Wang, H. Damages detection of aero-engine blades via deep learning algorithms. *IEEE Trans. Instrum. Meas.* 2023, 72, 5009111.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 39, 1137–1149.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
9. Jiao, L.C.; Zhang, F.; Liu, F.; Yang, S.Y.; Li, L.L.; Feng, Z.X.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* 2019, 7, 128837–128868.
10. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone Detection Network with Tiny Iterative Backbone. *IEEE Access* 2020, 8, 130697–130707.
11. He, J.; Liu, M.; Yu, C. UAV reaction detection based on multi-scale feature fusion. In Proceedings of the 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Xi'an, China, 28–30 October 2022; pp. 640–643.
12. Wastupranata, L.M.; Munir, R. UAV Detection using Web Application Approach based on SSD Pre-Trained Model. In Proceedings of the 2021 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Virtual, 3–4 November 2021; pp. 1–6.
13. Tao, Y.; Zongyang, Z.; Jun, Z.; Xinghua, C.; Fuqiang, Z. Low-altitude small-sized object detection using lightweight feature-enhanced convolutional neural network. *J. Syst. Eng. Electron.* 2021, 32, 841–853.
14. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An Efficient Network for Low-Altitude Object Detection Based on Convolution and Transformer. *IEEE Trans. Instrum. Meas.* 2022, 71, 2507412.
15. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* 2023, 111, 257–276.
16. Ma, J.; Yao, Z.; Xu, C.; Chen, S. Multi-UAV real-time tracking algorithm based on improved PP-YOLO and Deep-SORT. *J. Comput. Appl.* 2022, 42, 2885.
17. Li, H.; Yang, J.; Mao, Y.; Hu, Q.; Du, Y.; Peng, J.; Liu, C. A UAV detection algorithm combined with lightweight network. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; Volume 5, pp. 1865–1872.
18. Liu, Y.; Liu, D.; Wang, B.; Chen, B. Mob-YOLO: A Lightweight UAV Object Detection Method. In Proceedings of the 2022 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial

Intelligence (ICSMD), Harbin, China, 30 November–2 December 2022; pp. 1–6.

19. Liu, R.; Xiao, Y.; Li, Z.; Cao, H. Research on the anti-UAV distributed system for airports: YOLOv5-based auto-targeting device. In Proceedings of the 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), Changchun, China, 20–22 May 2022; pp. 864–867.

20. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* 2023, 7, 304.

21. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1904–1916.

22. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

24. Meng, Y., et al.: A swintransformer based bird nest detection approach with unmanned aerial vehicle images for power distribution and pole towers. *IET Cyber-Phys. Syst., Theory Appl.* 1–10 2023.

25. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* 2021, arXiv:2112.05561.

#### References

1. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893

2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 2004, 60, 91–110.

3. Bay, H.; Tuytelaars, T.; van Gool, L. Surf: Speeded up robust features. In Proceedings of the Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part I 9. Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.

4. Dai, J.; Wu, L.; Wang, P. Overview of UAV Target Detection Algorithms Based on Deep Learning. In Proceedings of the 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 17–19 December 2021; Volume 2, pp. 736–745.

5. Zuo, Y. Target Detection System of Agricultural Economic Output Efficiency Based on Kruskal Algorithm. In Proceedings of the 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNBC), Tumkur, India, 2–3 December 2022; pp. 1–5.

6. Li, S.; Yu, J.; Wang, H. Damages detection of aero-engine blades via deep learning algorithms. *IEEE Trans. Instrum. Meas.* 2023, 72, 5009111.

7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 39, 1137–1149.

8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

9. Jiao, L.C.; Zhang, F.; Liu, F.; Yang, S.Y.; Li, L.L.; Feng, Z.X.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* 2019, 7, 128837–128868.

10. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone Detection Network with Tiny Iterative Backbone. *IEEE Access* 2020, 8, 130697–130707.

11. He, J.; Liu, M.; Yu, C. UAV reaction detection based on multi-scale feature fusion. In Proceedings of the

2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Xi'an, China, 28–30 October 2022; pp. 640–643.

12. Wastupranata, L.M.; Munir, R. UAV Detection using Web Application Approach based on SSD Pre-Trained Model. In Proceedings of the 2021 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Virtual, 3–4 November 2021; pp. 1–6.

13. Tao, Y.; Zongyang, Z.; Jun, Z.; Xinghua, C.; Fuqiang, Z. Low-altitude small-sized object detection using lightweight feature-enhanced convolutional neural network. *J. Syst. Eng. Electron.* 2021, 32, 841–853.

14. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An Efficient Network for Low-Altitude Object Detection Based on Convolution and Transformer. *IEEE Trans. Instrum. Meas.* 2022, 71, 2507412.

15. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* 2023, 111, 257–276.

16. Ma, J.; Yao, Z.; Xu, C.; Chen, S. Multi-UAV real-time tracking algorithm based on improved PP-YOLO and Deep-SORT. *J. Comput. Appl.* 2022, 42, 2885.

17. Li, H.; Yang, J.; Mao, Y.; Hu, Q.; Du, Y.; Peng, J.; Liu, C. A UAV detection algorithm combined with lightweight network. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; Volume 5, pp. 1865–1872.

18. Liu, Y.; Liu, D.; Wang, B.; Chen, B. Mob-YOLO: A Lightweight UAV Object Detection Method. In Proceedings of the 2022 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMAD), Harbin, China, 30 November–2 December 2022; pp. 1–6.

19. Liu, R.; Xiao, Y.; Li, Z.; Cao, H. Research on the anti-UAV distributed system for airports: YOLOv5-based auto-targeting device. In Proceedings of the 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), Changchun, China, 20–22 May 2022; pp. 864–867.

20. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* 2023, 7, 304.

21. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1904–1916.

22. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

24. Sunkara, R.; Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. *arXiv* 2022, arXiv:2208.03641.

25. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* 2021, arXiv:2112.05561.

## Додаток В

### Програмні коди

Swin Transformer:

```
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
from .conv import Conv

class DropPath(nn.Module):
    """Drop paths (Stochastic Depth) per sample (when
    applied in main path of residual blocks).
    """
    def __init__(self, drop_prob=None):# mg
        super(DropPath, self).__init__()
        self.drop_prob = drop_prob

    def forward(self, x):
        return drop_path_f(x, self.drop_prob, self.training)

def drop_path_f(x, drop_prob: float = 0., training: bool =
False):
    """Drop paths (Stochastic Depth) per sample (when
    applied in main path of residual blocks).
    This is the same as the DropConnect impl I created for
    EfficientNet, etc networks, however,
    the original name is misleading as 'Drop Connect' is a
    different form of dropout in a separate paper...
    See discussion:
    https://github.com/tensorflow/tpu/issues/494#issuecomment-
    532968956 ... I've opted for
    changing the layer and argument names to 'drop path'
    rather than mix DropConnect as a layer name and use
    'survival rate' as the argument.
    """
    if drop_prob == 0. or not training:
        return x
    keep_prob = 1 - drop_prob
    shape = (x.shape[0],) + (1,) * (x.ndim - 1) # work with
diff dim tensors, not just 2D ConvNets
    random_tensor = keep_prob + torch.rand(shape,
dtype=x.dtype, device=x.device)
    random_tensor.floor_() # binarize
    output = x.div(keep_prob) * random_tensor
    return output

##### swin transformer #####
class WindowAttention(nn.Module):
```

```
def __init__(self, dim, window_size, num_heads,
qkv_bias=True, qk_scale=None, attn_drop=0., proj_drop=0.):

    super().__init__()
    self.dim = dim
    self.window_size = window_size # Wh, Ww
    self.num_heads = num_heads
    head_dim = dim // num_heads
    self.scale = qk_scale or head_dim ** -0.5

    # define a parameter table of relative position bias
    self.relative_position_bias_table = nn.Parameter(
        torch.zeros((2 * window_size[0] - 1) * (2 *
window_size[1] - 1), num_heads)) # 2*Wh-1 * 2*Ww-1, nH

    # get pair-wise relative position index for each
token inside the window
    coords_h = torch.arange(self.window_size[0])
    coords_w = torch.arange(self.window_size[1])
    coords = torch.stack(torch.meshgrid([coords_h,
coords_w])) # 2, Wh, Ww
    coords_flatten = torch.flatten(coords, 1) # 2,
Wh*Ww
    relative_coords = coords_flatten[:, :, None] -
coords_flatten[:, None, :] # 2, Wh*Ww, Wh*Ww
    relative_coords = relative_coords.permute(1, 2,
0).contiguous() # Wh*Ww, Wh*Ww, 2
    relative_coords[:, :, 0] += self.window_size[0] - 1
# shift to start from 0
    relative_coords[:, :, 1] += self.window_size[1] - 1
    relative_coords[:, :, 0] *= 2 * self.window_size[1]
- 1
    relative_position_index = relative_coords.sum(-1) #
Wh*Ww, Wh*Ww
    self.register_buffer("relative_position_index",
relative_position_index)

    self.qkv = nn.Linear(dim, dim * 3, bias=qkv_bias)
    self.attn_drop = nn.Dropout(attn_drop)
    self.proj = nn.Linear(dim, dim)
    self.proj_drop = nn.Dropout(proj_drop)

    nn.init.normal_(self.relative_position_bias_table,
std=.02)
    self.softmax = nn.Softmax(dim=-1)

    def forward(self, x, mask=None):

        B_, N, C = x.shape
        qkv = self.qkv(x).reshape(B_, N, 3, self.num_heads,
C // self.num_heads).permute(2, 0, 3, 1, 4)
```

```

    q, k, v = qkv[0], qkv[1], qkv[2] # make torchscript
    happy (cannot use tensor as tuple)

    q = q * self.scale
    attn = (q @ k.transpose(-2, -1))

    relative_position_bias =
self.relative_position_bias_table[self.relative_position_ind
ex.view(-1)].view(
    self.window_size[0] * self.window_size[1],
self.window_size[0] * self.window_size[1], -1) #
Wh*Ww,Wh*Ww,nH
    relative_position_bias =
relative_position_bias.permute(2, 0, 1).contiguous() # nH,
Wh*Ww, Wh*Ww
    attn = attn + relative_position_bias.unsqueeze(0)

    if mask is not None:
        nW = mask.shape[0]
        attn = attn.view(B_ // nW, nW, self.num_heads,
N, N) + mask.unsqueeze(1).unsqueeze(0)
        attn = attn.view(-1, self.num_heads, N, N)
        attn = self.softmax(attn)
    else:
        attn = self.softmax(attn)

    attn = self.attn_drop(attn)

    # print(attn.dtype, v.dtype)
    try:
        x = (attn @ v).transpose(1, 2).reshape(B_, N, C)
    except:
        # print(attn.dtype, v.dtype)
        x = (attn.half() @ v).transpose(1,
2).reshape(B_, N, C)
    x = self.proj(x)
    x = self.proj_drop(x)
    return x

class Mlp(nn.Module): # tc

    def __init__(self, in_features, hidden_features=None,
out_features=None, act_layer=nn.SiLU, drop=0.):
        super().__init__()
        out_features = out_features or in_features
        hidden_features = hidden_features or in_features
        self.fc1 = nn.Linear(in_features, hidden_features)
        self.act = act_layer()
        self.fc2 = nn.Linear(hidden_features, out_features)
        self.drop = nn.Dropout(drop)

    def forward(self, x):
        x = self.fc1(x)
        x = self.act(x)
        x = self.drop(x)

```

```

    x = self.fc2(x)
    x = self.drop(x)
    return x

def window_partition(x, window_size):
    B, H, W, C = x.shape
    assert H % window_size == 0, 'feature map h and w can
not divide by window size'
    x = x.view(B, H // window_size, window_size, W //
window_size, window_size, C)
    windows = x.permute(0, 1, 3, 2, 4,
5).contiguous().view(-1, window_size, window_size, C)
    return windows

def window_reverse(windows, window_size, H, W):
    B = int(windows.shape[0] / (H * W / window_size /
window_size))
    x = windows.view(B, H // window_size, W // window_size,
window_size, window_size, -1)
    x = x.permute(0, 1, 3, 2, 4, 5).contiguous().view(B, H,
W, -1)
    return x

class SwinTransformerLayer(nn.Module):

    def __init__(self, dim, num_heads, window_size=8,
shift_size=0,
                    mlp_ratio=4., qkv_bias=True, qk_scale=None,
drop=0., attn_drop=0., drop_path=0.,
                    act_layer=nn.SiLU,
norm_layer=nn.LayerNorm):
        super().__init__()
        self.dim = dim
        self.num_heads = num_heads
        self.window_size = window_size
        self.shift_size = shift_size
        self.mlp_ratio = mlp_ratio

        assert 0 <= self.shift_size < self.window_size,
"shift_size must in 0-window_size"

        self.norm1 = norm_layer(dim)
        self.attn = WindowAttention(
            dim, window_size=(self.window_size,
self.window_size), num_heads=num_heads,
            qkv_bias=qkv_bias, qk_scale=qk_scale,
            attn_drop=attn_drop, proj_drop=drop)

        self.drop_path = DropPath(drop_path) if drop_path >
0. else nn.Identity()
        self.norm2 = norm_layer(dim)
        mlp_hidden_dim = int(dim * mlp_ratio)

```

```

        self.mlp = Mlp(in_features=dim,
hidden_features=mlp_hidden_dim, act_layer=act_layer,
drop=drop)

    def create_mask(self, H, W):
        # calculate attention mask for SW-MSA
        img_mask = torch.zeros((1, H, W, 1)) # 1 H W 1
        h_slices = (slice(0, -self.window_size),
                    slice(-self.window_size, -
self.shift_size),
                    slice(-self.shift_size, None))
        w_slices = (slice(0, -self.window_size),
                    slice(-self.window_size, -
self.shift_size),
                    slice(-self.shift_size, None))
        cnt = 0
        for h in h_slices:
            for w in w_slices:
                img_mask[:, h, w, :] = cnt
                cnt += 1

        mask_windows = window_partition(img_mask,
self.window_size) # nW, window_size, window_size, 1
        mask_windows = mask_windows.view(-1,
self.window_size * self.window_size)
        attn_mask = mask_windows.unsqueeze(1) -
mask_windows.unsqueeze(2)
        attn_mask = attn_mask.masked_fill(attn_mask != 0,
float(-100.0)).masked_fill(attn_mask == 0, float(0.0))

        return attn_mask

    def forward(self, x):
        # reshape x[b c h w] to x[b l c]
        _, _, H_, W_ = x.shape

        Padding = False
        if min(H_, W_) < self.window_size or H_ %
self.window_size != 0 or W_ % self.window_size != 0:
            Padding = True
            # print(f'img_size {min(H_, W_)} is less than
(or not divided by) window_size {self.window_size},
Padding.')
            pad_r = (self.window_size - W_ %
self.window_size) % self.window_size
            pad_b = (self.window_size - H_ %
self.window_size) % self.window_size
            x = F.pad(x, (0, pad_r, 0, pad_b))

        # print('2', x.shape)
        B, C, H, W = x.shape
        L = H * W
        x = x.permute(0, 2, 3, 1).contiguous().view(B, L, C)
        # b, L, c

        # create mask from init to forward

        if self.shift_size > 0:
            attn_mask = self.create_mask(H, W).to(x.device)
        else:
            attn_mask = None

        shortcut = x
        x = self.norm1(x)
        x = x.view(B, H, W, C)

        # cyclic shift
        if self.shift_size > 0:
            shifted_x = torch.roll(x, shifts=(-
self.shift_size, -self.shift_size), dims=(1, 2))
        else:
            shifted_x = x

        # partition windows
        x_windows = window_partition(shifted_x,
self.window_size) # nW*B, window_size, window_size, C
        x_windows = x_windows.view(-1, self.window_size *
self.window_size, C) # nW*B, window_size*window_size, C

        # W-MSA/SW-MSA
        attn_windows = self.attn(x_windows, mask=attn_mask)
        # nW*B, window_size*window_size, C

        # merge windows
        attn_windows = attn_windows.view(-1,
self.window_size, self.window_size, C)
        shifted_x = window_reverse(attn_windows,
self.window_size, H, W) # B H' W' C

        # reverse cyclic shift
        if self.shift_size > 0:
            x = torch.roll(shifted_x,
shifts=(self.shift_size, self.shift_size), dims=(1, 2))
        else:
            x = shifted_x
        x = x.view(B, H * W, C)

        # FFN
        x = shortcut + self.drop_path(x)
        x = x + self.drop_path(self.mlp(self.norm2(x)))

        x = x.permute(0, 2, 1).contiguous().view(-1, C, H,
W) # b c h w

        if Padding:
            x = x[:, :, :H_, :W_] # reverse padding

        return x

class SwinTransformerBlock(nn.Module):
    def __init__(self, c1, c2, num_heads, num_layers,
window_size=8):

```

```

    super().__init__()
    self.conv = None
    if c1 != c2:
        self.conv = Conv(c1, c2)

    # remove input_resolution
    self.blocks =
nn.Sequential(*[SwinTransformerLayer(dim=c2,
num_heads=num_heads, window_size=window_size,
shift_size=0 if (i % 2 == 0) else window_size // 2) for i in
                range(num_layers)])

def forward(self, x):
    if self.conv is not None:
        x = self.conv(x)
    x = self.blocks(x)
    return x

class C3STR(C3):
    # C3 module with SwinTransformerBlock()
    def __init__(self, c1, c2, n=1, shortcut=True, g=1,
e=0.5):
        super().__init__(c1, c2, n, shortcut, g, e)
        c_ = int(c2 * e)
        num_heads = c_ // 32
        self.m = SwinTransformerBlock(c_, c_, num_heads, n)

class STCSPA(nn.Module):
    # CSP Bottleneck
https://github.com/WongKinYiu/CrossStagePartialNetworks
    def __init__(self, c1, c2, n=1, shortcut=True, g=1,
e=0.5): # ch_in, ch_out, number, shortcut, groups,
expansion
        super(STCSPA, self).__init__()
        c_ = int(c2 * e) # hidden channels
        self.cv1 = Conv(c1, c_, 1, 1)
        self.cv2 = Conv(c1, c_, 1, 1)
        self.cv3 = Conv(2 * c_, c2, 1, 1)
        num_heads = c_ // 32
        self.m = SwinTransformerBlock(c_, c_, num_heads, n)
        # self.m = nn.Sequential(*[Bottleneck(c_, c_,
shortcut, g, e=1.0) for _ in range(n)])

    def forward(self, x):
        y1 = self.m(self.cv1(x))
        y2 = self.cv2(x)
        return self.cv3(torch.cat((y1, y2), dim=1))

class STCSPB(nn.Module):
    # CSP Bottleneck
https://github.com/WongKinYiu/CrossStagePartialNetworks
    def __init__(self, c1, c2, n=1, shortcut=False, g=1,
e=0.5): # ch_in, ch_out, number, shortcut, groups,
expansion
        super(STCSPB, self).__init__()
        c_ = int(c2) # hidden channels
        self.cv1 = Conv(c1, c_, 1, 1)
        self.cv2 = Conv(c_, c_, 1, 1)
        self.cv3 = Conv(2 * c_, c2, 1, 1)
        num_heads = c_ // 32
        self.m = SwinTransformerBlock(c_, c_, num_heads, n)
        # self.m = nn.Sequential(*[Bottleneck(c_, c_,
shortcut, g, e=1.0) for _ in range(n)])

    def forward(self, x):
        x1 = self.cv1(x)
        y1 = self.m(x1)
        y2 = self.cv2(x1)
        return self.cv3(torch.cat((y1, y2), dim=1))

class STCSPC(nn.Module):
    # CSP Bottleneck
https://github.com/WongKinYiu/CrossStagePartialNetworks
    def __init__(self, c1, c2, n=1, shortcut=True, g=1,
e=0.5): # ch_in, ch_out, number, shortcut, groups,
expansion
        super(STCSPC, self).__init__()
        c_ = int(c2 * e) # hidden channels
        self.cv1 = Conv(c1, c_, 1, 1)
        self.cv2 = Conv(c1, c_, 1, 1)
        self.cv3 = Conv(c_, c_, 1, 1)
        self.cv4 = Conv(2 * c_, c2, 1, 1)
        num_heads = c_ // 32
        self.m = SwinTransformerBlock(c_, c_, num_heads, n)
        # self.m = nn.Sequential(*[Bottleneck(c_, c_,
shortcut, g, e=1.0) for _ in range(n)])

    def forward(self, x):
        y1 = self.cv3(self.m(self.cv1(x)))
        y2 = self.cv2(x)
        return self.cv4(torch.cat((y1, y2), dim=1))

GAM:

class GAM(nn.Module):
    def __init__(self, c1, c2, group=True, rate=4):
        super(GAM, self).__init__()

        self.channel_attention = nn.Sequential(
            nn.Linear(c1, int(c1 / rate)),
            nn.ReLU(inplace=True),
            nn.Linear(int(c1 / rate), c1)
        )

        self.spatial_attention = nn.Sequential(
            nn.Conv2d(c1, c1 // rate, kernel_size=7,
padding=3, groups=rate) if group else nn.Conv2d(c1, int(c1 /
rate),

```

```

kernel_size=7,

padding=3),
    nn.BatchNorm2d(int(c1 / rate)),
    nn.ReLU(inplace=True),
    nn.Conv2d(c1 // rate, c2, kernel_size=7,
padding=3, groups=rate) if group else nn.Conv2d(int(c1 /
rate), c2,

kernel_size=7,

padding=3),
    nn.BatchNorm2d(c2)
)

def forward(self, x):
    b, c, h, w = x.shape
    x_permute = x.permute(0, 2, 3, 1).view(b, -1, c)
    x_att_permute =
self.channel_attention(x_permute).view(b, h, w, c)
    x_channel_att = x_att_permute.permute(0, 3, 1, 2)
    # x_channel_att=channel_shuffle(x_channel_att,4)
#last shuffle
    x = x * x_channel_att

    x_spatial_att = self.spatial_attention(x).sigmoid()
    x_spatial_att = channel_shuffle(x_spatial_att, 4) #
last shuffle
    out = x * x_spatial_att
    # out=channel_shuffle(out,4) #last shuffle
    return out

Model config:

# Ultralytics YOLO 🦉, AGPL-3.0 license
# YOLOv8 object detection model with P2-P5 outputs. For
Usage examples see https://docs.ultralytics.com/tasks/detect

# Parameters
nc: 1 # number of classes
scales: # model compound scaling constants, i.e.
'model=yolov8n.yaml' will call yolov8.yaml with scale 'n'
# [depth, width, max_channels]
n: [0.33, 0.25, 1024]
s: [0.33, 0.50, 1024]
m: [0.67, 0.75, 768]
l: [1.00, 1.00, 512]

```

```

x: [1.00, 1.25, 512]

# YOLOv8.0 backbone
backbone:
# [from, repeats, module, args]
- [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
- [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
- [-1, 3, C2f, [128, True]]
- [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
- [-1, 6, C2f, [256, True]]
- [-1, 1, Conv, [512, 3, 2]] # 5-P4/16
- [-1, 9, C3STR, [512]]
- [-1, 1, Conv, [1024, 3, 2]] # 7-P5/32
- [-1, 3, C3STR, [1024]]
- [-1, 1, SPPF, [1024, 5]] # 9

# YOLOv8.0-p2 head
head:
- [-1, 1, nn.Upsample, [None, 2, 'nearest']]
- [[-1, 6], 1, Concat, [1]] # cat backbone P4
- [-1, 3, C2f, [512]] # 12

- [-1, 1, nn.Upsample, [None, 2, 'nearest']]
- [[-1, 4], 1, Concat, [1]] # cat backbone P3
- [-1, 3, C2f, [256]] # 15 (P3/8-small)
- [-1, 3, GAM, [256]]

- [-1, 1, nn.Upsample, [None, 2, 'nearest']]
- [[-1, 2], 1, Concat, [1]] # cat backbone P2
- [-1, 3, C2f, [128]] # 19 (P2/4-xsmall)
- [-1, 3, GAM, [128]]

- [-1, 1, Conv, [128, 3, 2]]
- [[-1, 15], 1, Concat, [1]] # cat head P3
- [-1, 3, C2f, [256]] # 23 (P3/8-small)
- [-1, 3, GAM, [256]]

- [-1, 1, Conv, [256, 3, 2]]
- [[-1, 12], 1, Concat, [1]] # cat head P4
- [-1, 3, C2f, [512]] # 27 (P4/16-medium)
- [-1, 3, GAM, [512]]

- [[19, 23, 27], 1, Detect, [nc]] # Detect(P2, P3, P4)

```



**ВІДГУК НАУКОВОГО КЕРІВНИКА  
на кваліфікаційну роботу магістра**

*гр. КНм-22-1 Горлова Сергія Вікторовича за темою: Метод виявлення багатороторних БПЛА засобами штучного інтелекту*

**1. Актуальність теми**

З огляду на зростаючу популярність і доступність БПЛА, важливість розробки надійних систем виявлення є важливою та актуальною. Робота відображає актуальність теми, акцентуючи на значущості виявлення БПЛА у різних сферах за допомогою штучного інтелекту, особливо у контексті забезпечення безпеки повітряного простору.

**2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт**

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу у сфері комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей глибокого навчання) до розпізнавання зображень обличчя людей під різними кутами зору, та при різній освітленості).

**3. Професійні та особистісні якості магістранта**

Магістрант продемонстрував ретельний підхід до аналізу теми. Його здібності до аналізу та розуміння комплексних технічних питань стали очевидними в ході роботи. Він показав себе як зосереджену на цілях особистість, здатну ефективно працювати у різних умовах. На особистісному рівні магістрант показав себе як віддану, цілеспрямовану та мотивовану особистість.

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

Основну частину дослідження, включно з розробкою методу виявлення багатороторних БПЛА засобами штучного інтелекту, виконано особисто. Це охоплює збір та аналіз даних, розробку алгоритмів та їх тестування. Внесок магістранта також включав самостійну розробку та налаштування програмного забезпечення для проведення експериментів. Він виявив високий рівень ініціативи та самостійності у вирішенні проблем, що виникали в процесі дослідження.

**5. Наукова новизна та оригінальність запропонованих підходів**

Робота включає інноваційні елементи у вигляді розробки та адаптації алгоритмів, що є новаторськими для даної сфери. Це відображає високий рівень оригінальності та творчого мислення магістранта.

Однак, у роботі може бути недостатньо чітко визначено межі наукової новизни, що може ускладнити її сприйняття та оцінку. Важливо більш детально окреслити, як саме запропоновані підходи відрізняються від існуючих методів та який внесок вони роблять у наукову сферу.

Результати дослідження опубліковані у віснику Хмельницького національного університету.

#### **6. Ступінь оволодіння методами дослідження**

Магістрант продемонстрував глибоке розуміння та ефективне застосування методів дослідження у своїй роботі. Використання сучасних технік штучного інтелекту та алгоритмічних підходів свідчить про високий ступінь оволодіння необхідними методами.

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

Тема роботи розкрита повністю та якісно. Мета дослідження була чітко визначена, а зміст роботи відображає всебічне та детальне дослідження обраної тематики.

#### **8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу**

Виклад матеріалу у кваліфікаційній роботі магістра характеризується високою логічністю та послідовністю. Презентація даних та аргументація відбувається в зрозумілій, лаконічній формі. Матеріал подано таким чином, що легко просліджувати розвиток ідей від вступу до висновків. Літературна та граматична якість тексту висока, що свідчить про уважність магістранта до деталей та важливість якості презентації матеріалу.

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

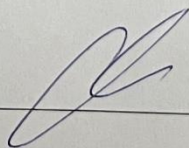
Розроблені методи та підходи можуть бути застосовані у реальних умовах для виявлення та класифікації БПЛА, що має потенціал для використання в оборонній сфері, безпеці та контролю повітряного простору. Практичне застосування цих методів може значно вплинути на покращення існуючих систем безпеки.

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

На основі викладених вище пунктів, рекомендую допустити кваліфікаційну роботу магістра до захисту. Робота відповідає всім необхідним критеріям, включаючи наукову новизну, логічність викладу, та має значний практичний потенціал.

Кваліфікаційна робота заслуговує на оцінку «Відмінно».

Науковий керівник \_\_\_\_\_ д.т.н., проф. Олександр Бармак





## ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

*гр. КВ№-22-1 Терлюва Сергія Валерійовича за темою: Методи виявлення балістичних БПЛА засобами штучного інтелекту*

### 1. Актуальність обраної теми

Враховуючи швидкий розвиток технологій безпілотних літальних апаратів та їх застосування у різних сферах, включаючи військові, цивільні та комерційні потреби, важливо розробити ефективні методи для їх виявлення та класифікації. Використання штучного інтелекту для виявлення повітряних цілей дозволить значно підвищити безпеку у повітряному просторі. Тому робота, виконана автором є актуальною та перспективною.

### 2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу з сфери комп'ютерного зору, яка є складовою предметної області комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей глибокого навчання) до розпізнавання зображень обличчя людей під різними кутами зору, та при різній освітленості). У роботі присутнє дослідження яке розвиває існуючі підходи та отримані нові результати які за певними показниками – вище, чим у аналогічних підходах.

### 3. Повнота розкриття мети та завдань дослідження

Автор проаналізував існуючі методи виявлення БПЛА, які використовують глибоке навчання, та успішно розробив вдосконалену модель для їхнього виявлення. Завдання дослідження виконано повністю та на високому рівні.

### 4. Наявність наукової новизни

Робота містить наукову новизну, представлену у формі розробки унікального методу виявлення БПЛА, що включає інтеграцію нових технологій в архітектуру YOLOv8. Продемонстровано й обґрунтовано результати, які мають наукове та інноваційне значення. Результати дослідження опубліковані у віснику Хмельницького національного університету.

### 5. Зміст кожного розділу роботи

Кваліфікаційна робота магістра складається з чотирьох розділів. Перший розділ зосереджується на огляді підходів до виявлення БПЛА за допомогою штучного інтелекту,

включаючи класифікацію БПЛА та аналіз існуючих методів і публікацій у цій галузі. Другий розділ присвячений методам виявлення багатороторних БПЛА, особливостям моделі глибокого навчання YOLOv8, покращенню здатності моделі до виявлення малих цілей, а також вдосконаленню архітектури мережі. Третій розділ описує проектування структури інформаційної системи для експериментального дослідження запропонованого методу, включаючи використання Python, пакету Ultralytics для YOLOv8 та платформи машинного навчання PyTorch. У четвертому розділі виконано аналіз набору даних для тестування дослідження, представлено експериментальні результати та проведено опис методів проведення експериментів.

#### **6. Ступінь розкриття теми роботи**

Тема роботи повністю розкрита, забезпечуючи глибоке розуміння проблематики та ефективні підходи до її розв'язання.

#### **7. Якість оформлення кваліфікаційної роботи**

Оформлення роботи відповідає необхідним нормам та вимогам, які ставляться до оформлення кваліфікаційних робіт.

#### **8. Недоліки кваліфікаційної роботи**

Суттєвих недоліків у роботі виявлено не було, проте додаткове дослідження та валідація запропонованих методів на більш розширеному наборі даних були б корисними.

#### **9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота**

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

Опонент к. ер.-м.н., доктор Змицька Н. О.



**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА  
РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ**

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

Назва: Метод виявлення багатороторних БПЛА засобами штучного інтелекту

Автор: С.В. Горелов

Спеціальність: 122 – Компютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., професор, зав. кафедри КН О.В.Бармак

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) за програмою Anti-Plagiarism виявлені 1% запозичень вказують на джерела посилань та відомі терміни.
- 2) За програмою UNICHECK виявлені 13.4% є фрагментарними – містять поширені конструкції та схеми нейромереж (у роботі вказані посилання на ці джерела), загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1% і 13.4% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

О. В. Бармак

Р.  Багрій

О. В. Бармак

Ім'я користувача:  
Кафедра КН

ID перевірки:  
1016000768

Дата перевірки:  
13.12.2023 11:17:51 EET

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
13.12.2023 11:19:48 EET

ID користувача:  
100005671

Назва документа: КНм-22-1 Горелов

Кількість сторінок: 85 Кількість слів: 14391 Кількість символів: 106926 Розмір файлу: 4.59 MB ID файлу: 1015684337

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

## 13.4% Схожість

Найбільша схожість: 3.25% з Інтернет-джерелом (<https://www.mdpi.com/2079-9292/12/17/3664>)

12.2% Джерела з Інтернету

819

Сторінка 87

3.18% Джерела з Бібліотеки

78

Сторінка 94

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Підозріле форматування

13  
сторінок

# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилоч в документах: 11%**

ID: 122909 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод виявлення багатороторних БПЛА засобами штучного інтелекту Додано в БД: 2023-12-13 Автора: С.В. Горелов Керівники: О.В. Бармак Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	87601	1321	1629 (2%)	27 (2%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми