

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи виявлення дронів за допомогою
глибокого навчання

(тема)

Виконав:

здобувач 2 року навчання,

групи СПМ-23-4

Євгенія ПАХОМОВА

(власне ім'я, прізвище)

Спеціальність

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування

(повна назва освітньої програми)

Керівник: доц. Віталій МАРТОВИЦЬКИЙ

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Пахомовій Євгенії Володимирівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи виявлення дронів за допомогою глибокого навчання _____

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 16 червня 2025 р.

3. Вхідні дані до роботи _____ Датасет зображень _____

4. Перелік питань, що потрібно опрацювати у роботі _____

Актуальність проблеми виявлення БПЛА (дронів) _____

Огляд існуючих методів виявлення дронів _____

Оптичне виявлення дронів за допомогою глибокого навчання _____

Типи моделей глибокого навчання для детекції об'єктів _____

Особливості виявлення малих об'єктів _____

Огляд моделей YOLO (v3, v4, v5) _____

Підготовка та анотація датасетів _____

Експерименти з різними датасетами та камерами _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 19 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|----------------------|--|---|------|
| | | підпис | дата |
| | | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Строк / терміни виконання етапів роботи | Примітка |
|---|--|---|----------|
| 1 | Огляд методів виявлення безпілотних апаратів | 22.04.25-29.04.25 | |
| 2 | Вибір та обґрунтування методики дослідження | 30.04.25-05.05.25 | |
| 3 | Вибір інструментальних засобів | 06.05.25-09.05.25 | |
| 4 | Розробка моделей | 10.05.25-20.05.25 | |
| 5 | Проведення експериментів | 21.05.25-02.06.25 | |
| 6 | Оформлення матеріалів кваліфікаційної роботи | 03.06.25-05.06.25 | |
| 7 | Подання кваліфікаційної роботи керівникові та її попередній захист | 06.06.25-09.06.25 | |
| 8 | Подання кваліфікаційної роботи на рецензування | 10.06.25-12.06.25 | |
| | | | |

Дата видачі завдання “ 21 ” квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Віталій МАРТОВИЦЬКИЙ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 57 с., 16 рис., 6 табл., 1 дод., 18 джерел.

БЕЗПЛОТНІ ЛІТАЛЬНІ АПАРАТИ, ГЛИБОКЕ НАВЧАННЯ, YOLO, ВИЯВЛЕННЯ ОБ'ЄКТІВ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ, РОЗБИТТЯ ЗОБРАЖЕНЬ (TILING), ТОЧНІСТЬ, ПОВНОТА, РЕАЛЬНИЙ ЧАС.

Метою кваліфікаційної роботи є розробка та дослідження методів виявлення безпілотних літальних апаратів (БПЛА) з використанням алгоритмів глибокого навчання для забезпечення ефективної та точної детекції дронів у режимі реального часу.

У ході виконання кваліфікаційної роботи було проведено огляд сучасних підходів до виявлення дронів, зокрема оптичних методів на основі комп'ютерного зору. Основну увагу приділено аналізу та порівнянню архітектур моделей YOLO (v3, v4, v5), які є одноетапними детекторами об'єктів, здатними забезпечити швидке виявлення візуальних об'єктів. Досліджено проблему детекції малих об'єктів, характерну для задач виявлення дронів на великій відстані, та застосовано метод розбиття зображень на фрагменти (tiling) для покращення точності. Проведено серію експериментів із використанням як реальних, так і симульованих даних, у тому числі зображень дронів і птахів, для зменшення кількості хибнопозитивних спрацювань. Ефективність моделей оцінювалась за метриками точності, повноти та часу інференсу. За результатами досліджень сформульовано рекомендації щодо вибору архітектури моделі в залежності від обмежень апаратного забезпечення та цілей застосування.

ABSTRACT

Master's thesis: 57 pages, 16 figures, 6 tables, 1 appendices, 18 sources.

UNMANNED AERIAL VEHICLES, DEEP LEARNING, YOLO, OBJECT DETECTION, COMPUTER VISION, NEURAL NETWORKS, IMAGE TILING, PRECISION, RECALL, REAL TIME.

The major goal of this thesis is the development and research of methods for unmanned aerial vehicle (UAV) detection using deep learning algorithms to ensure efficient and accurate drone detection in real time. During the course of this qualification work, a review of current approaches to drone detection was conducted, particularly optical methods based on computer vision. The main focus was placed on analyzing and comparing YOLO architectures (v3, v4, v5), which are one-stage object detectors capable of fast visual object recognition. The problem of detecting small objects, which is typical in drone detection at long distances, was studied, and the method of image tiling was applied to improve accuracy. A series of experiments was conducted using both real and simulated data, including drone and bird images, to reduce the number of false positives. The models' performance was evaluated using precision, recall, and inference time metrics. Based on the findings, recommendations were formulated regarding model architecture selection depending on hardware limitations and application goals.

ЗМІСТ

| | |
|--|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ | 7 |
| ВСТУП | 8 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ | 9 |
| 1.1 Аналіз сучасного стану предметної області..... | 9 |
| 1.2 Постановка проблеми | 10 |
| 2 МЕТОДИ ВИЯВЛЕННЯ БПЛА..... | 12 |
| 2.1 виявлення безпілотників..... | 12 |
| 2.2 Глибоке навчання у виявленні об'єктів..... | 13 |
| 2.3 виявлення малих об'єктів..... | 17 |
| 3 ВИЯВЛЕННЯ ДРОНІВ ЗА ДОПОМОГОЮ ГЛИБОКОГО НАВЧАННЯ..... | 19 |
| 3.1 Огляд вхідних даних | 19 |
| 3.2 Розбиття зображень на фрагменти | 20 |
| 3.3 Метрики оцінювання | 24 |
| 3.4 Статистичний аналіз | 27 |
| 4 ОПИС ЕКСПЕРЕМЕНТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ | 28 |
| 4.1 Опис експерименту | 28 |
| 4.2 Вибір моделі SOTA..... | 32 |
| ВИСНОВКИ..... | 42 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 44 |
| ДОДАТОК А Графічний матеріал кваліфікаційної роботи..... | 47 |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- БПЛА – безпілотний літальний апарат
- ГН – глибоке навчання
- НМ – нейронна мережа
- КЗ – комп’ютерний зір
- ОК – обмежувальна коробка (bounding box)
- ПП – передобробка зображень
- ТТА – тестова аугментація (англ., Test-Time Augmentation)
- FPS – кількість кадрів за секунду (англ., Frames Per Second)
- IoU – коефіцієнт перетину з об’єднанням (англ., Intersection over Union)
- TP – правильне позитивне спрацювання (англ., True Positive)
- FP – хибне позитивне спрацювання (англ., False Positive)
- FN – хибне негативне спрацювання (англ., False Negative)
- TN – правильне негативне спрацювання (англ., True Negative)
- P – точність (англ., Precision)
- R – повнота (англ., Recall)
- YOLO – алгоритм виявлення об’єктів (англ., You Only Look Once)
- CNN – згорткова нейронна мережа (англ., Convolutional Neural Network)
- RPN – мережа пропозицій регіонів (англ., Region Proposal Network)

ВСТУП

У сучасному світі стрімкий розвиток безпілотних літальних апаратів (БПЛА), зокрема дронів, відкриває широкі можливості для комерційного, наукового та побутового використання. Разом із тим, їх зростаюча доступність та функціональність породжує нові виклики у сфері безпеки, приватності та контролю повітряного простору. Зокрема, дрони можуть бути використані для незаконного спостереження, доставки заборонених предметів або навіть скоєння диверсій.

У зв'язку з цим постає актуальне завдання – ефективно виявлення та ідентифікація дронів у реальному часі. Традиційні методи, такі як радарне спостереження або акустичний аналіз, мають свої обмеження, особливо у складних умовах або при наявності великої кількості об'єктів у повітряному просторі. Водночас, стрімкий розвиток технологій штучного інтелекту, зокрема глибокого навчання, відкриває нові перспективи для створення більш точних, гнучких та ефективних систем виявлення дронів.

Метою цієї кваліфікаційної роботи є дослідження сучасних методів виявлення дронів із використанням глибокого навчання, аналіз існуючих архітектур нейронних мереж, побудова та навчання моделей, а також оцінка їх ефективності у вирішенні поставленої задачі. У роботі розглядаються як теоретичні аспекти, так і практичні результати реалізації обраного підходу.

Актуальність теми визначається потребою у розробці інтелектуальних систем моніторингу повітряного простору, які могли б адаптуватися до нових загроз і працювати в режимі реального часу з високим рівнем точності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз сучасного стану предметної області

З розвитком технологій безпілотні літальні апарати (БПЛА), зокрема дрони, стали доступними та масово використовуються як у цивільних, так і у військових цілях. З одного боку, це сприяє розвитку нових сервісів – від аерозйомки до доставки товарів, а з іншого – створює ризики, пов'язані з безпекою, вторгненням у приватний простір, терористичними загрозами та незаконним збором інформації.

Традиційні методи виявлення дронів базуються на таких технологіях, як:

- радарні системи – ефективні для великих об'єктів, але мають труднощі з виявленням малих або низько летючих дронів;
- акустичні сенсори – можуть виявити характерний звук двигунів дронів, однак зазнають впливу шуму навколишнього середовища;
- інфрачервоні (тепловізійні) камери – здатні виявляти теплооб'єкти, проте ефективність знижується при складних погодних умовах;
- аналіз радіочастот – дозволяє визначати дрон за сигнатурою його зв'язку з оператором або gps, але вимагає специфічних технічних рішень.

Останніми роками все більшої популярності набувають методи комп'ютерного зору, зокрема ті, що базуються на глибокому навчанні. Вони дозволяють реалізовувати високоточне виявлення дронів на зображеннях або відео в режимі реального часу. Такі методи включають:

- Convolutional Neural Networks (CNN) – глибокі згорткові нейронні мережі, що ефективно виділяють просторові ознаки на зображеннях;
- YOLO (You Only Look Once) - сімейство алгоритмів, орієнтованих на швидке та точне виявлення об'єктів у реальному часі;

- SSD (Single Shot MultiBox Detector) та Faster R-CNN – популярні архітектури для задач детекції об'єктів;
- Transformer-based моделі (наприклад, DETR) – новітній підхід, що використовує механізм уваги для глобального аналізу зображення.

Незважаючи на значний прогрес, основні виклики у цій сфері залишаються:

- брак якісних та різноманітних датасетів для навчання моделей;
- складність виявлення дронів у складних умовах – при слабкому освітленні, на фоні дерев або будівель;
- висока потреба в оптимізації моделей для використання в реальному часі на вбудованих системах.

Таким чином, використання глибокого навчання у виявленні дронів є перспективним напрямом, що потребує подальшого дослідження, зокрема у частині збору спеціалізованих даних, адаптації моделей під реальні умови та інтеграції з іншими сенсорними системами, як-от радіочастотними або тепловізійними.

1.2 Постановка проблеми

Щоб відповідати технічним вимогам до камери, модель глибокого навчання повинна працювати достатньо швидко, щоб забезпечити виявлення дронів у режимі реального часу. Це пов'язано з тим, що дрони можуть швидко переміщуватися в кадрі, і навіть коротка затримка в обробці може призвести до втрати об'єкта спостереження. Таким чином, швидкість роботи моделі є критичним фактором для її практичного використання в системі моніторингу повітряного простору.

Крім того, для досягнення високої ефективності виявлення необхідно, щоб модель могла розпізнавати дрони ще на ранніх етапах їх появи в полі зору - тобто на великій відстані від камери. У таких випадках дрон займає дуже малу кількість пікселів на зображенні, що ускладнює його виявлення

навіть для найсучасніших алгоритмів. Це ставить перед дослідниками завдання створення або адаптації моделей, здатних з високою точністю виявляти дуже малі об'єкти.

У межах цієї кваліфікаційної роботи буде розглянуто низку дослідницьких питань, які мають на меті покращити якість та надійність виявлення дронів за допомогою методів глибокого навчання:

- які метрики є найдоцільнішими для оцінки якості виявлення дронів? сюди можуть входити такі показники, як точність (precision), повнота (recall), середнє значення точності (map), швидкість обробки кадрів (fps) тощо;

- яка з наявних сучасних моделей (state-of-the-art, sota) найкраще підходить для задачі виявлення дронів у реальному часі? буде проведено огляд і порівняння популярних архітектур, таких як yolo, ssd, faster r-cnn, detr та інших;

- які методи або підходи можуть бути використані для підвищення точності виявлення малих дронів, зокрема коли вони займають дуже незначну частину зображення;

- як зменшити кількість хибних спрацьовувань, що можуть виникати внаслідок схожості між дронами та птахами? це питання особливо актуальне для систем, що працюють у відкритому середовищі з великою кількістю фонових об'єктів.

Вирішення цих питань дозволить покращити точність, швидкість і надійність систем виявлення дронів, що базуються на глибокому навчанні, та сприятиме подальшому розвитку технологій захисту повітряного простору.

2 МЕТОДИ ВИЯВЛЕННЯ БПЛА

2.1 Виявлення безпілотників

Промислові та академічні підходи до виявлення дронів включають неоптичні методи, такі як використання акустики [1], радарів [2] та радіочастотного аналізу [3], а також оптичні методи, які здійснюють розпізнавання дронів на основі ознак, отриманих із фотографій та відеозаписів [4] [5].

Акустичні характеристики роторів дронів можуть використовуватись для їх розрізнення від звуків навколишнього середовища [1]. Різні звуки, включаючи дрони, літаки, птахів та грози, збираються мікрофонними масивами та передаються до алгоритмів, які визначають наявність дрона на основі високочастотних ознак [1]. Проте ця технологія є малоефективною в шумних місцях, таких як аеропорти та міські райони [6].

Радари широко використовуються для виявлення великих повітряних об'єктів, таких як літаки. Доплерівські радари також модифікуються для виявлення частотних зсувів від роторів і рухів крил дронів [2], хоча такі модифікації можуть створювати додаткові шуми і ускладнювати розрізнення між дронами, птахами та фоновими перешкодами [7].

Дрони використовують радіочастотні (RF) сигнали для зв'язку з наземним оператором і передачі відео на пульт керування. Ці RF-сигнали можна перехопити та проаналізувати за допомогою таких методів, як машинне навчання [8]. Виявлення радіочастотних сигналів є ефективним і може виконуватись на великих відстанях, що робить цей підхід одним із найпоширеніших на ринку технологій виявлення дронів [7].

У порівнянні з раніше розглянутими методами, оптичні підходи вважаються зручнішими, інтуїтивно зрозумілими та економічно вигідними.

Оптичні методи можна поділити на дві категорії залежно від того, як саме вони здійснюють вилучення ознак для виявлення та класифікації об'єктів.

Традиційні методи виявлення об'єктів базуються на ручному вилученні ознак та алгоритмах машинного навчання [9]. У таких підходах спочатку використовуються методи на зразок віднімання фону для виявлення області інтересу [10]. У більшості таких застосувань камери є квазістатичними, тобто або майже не рухаються, або взагалі нерухомі, і тільки об'єкти переміщуються у відеопотоці [10]. Далі вилучаються дескриптори ознак, такі як гістограма напрямлених градієнтів (HOG) [11], фур'є-дескриптори [12] та Speeded Up Robust Features (SURF) [13], які потім подаються на вхід класифікаторам для розпізнавання. Такі методи зазвичай працюють швидше, але мають нижчу точність розпізнавання порівняно з методами глибокого навчання.

Глибоке навчання нині широко використовується у задачах виявлення об'єктів. Вилучення ознак у таких підходах зазвичай здійснюється за допомогою згорткових шарів, які здатні виявляти не лише базові піксельні ознаки, а й високорівневі семантичні характеристики зображень. У результаті методи глибокого навчання формують більш якісні ієрархічні ознаки. Багато сучасних моделей (SOTA) уже були протестовані та використані у задачах виявлення дронів. Одним із завдань цієї роботи є тестування та порівняння нових SOTA-моделей виявлення об'єктів відповідно до вимог до системи виявлення дронів.

2.2 Глибоке навчання у виявленні об'єктів

У галузі комп'ютерного зору існує три основні задачі: класифікація, виявлення та сегментація. Мета роботи - побудова рамки (bounding box), що обмежує дрон на кадрі, вимагає інформації як про те, що зображено, так і де це розташоване, тому ця задача належить до категорії виявлення об'єктів.

У глибокому навчанні для задачі виявлення об'єктів існує два основних типи моделей: двохетапні детектори та одноетапні детектори.

Двохетапні детектори складаються з двох основних етапів. Перший етап відповідає за генерацію набору регіонів, які потенційно можуть містити об'єкти. Другий етап класифікує ці регіони та формує рамки навколо виявлених об'єктів. Типовими прикладами сучасних моделей двохетапного типу є RCNN (Region-Based Convolutional Neural Networks) [14], Fast RCNN [15] та Faster RCNN [15].

Структура Faster RCNN зображена на рисунку 2.1.

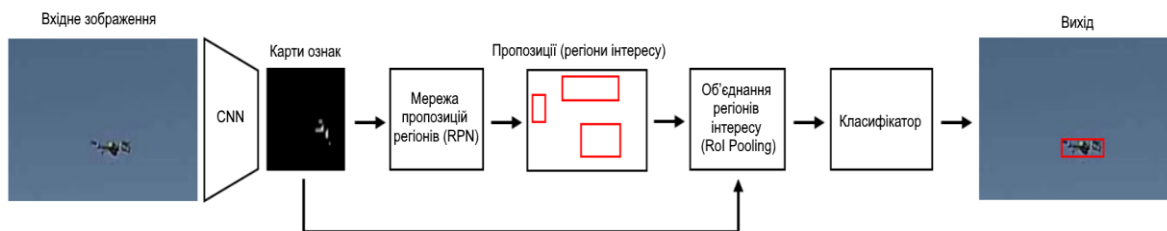


Рисунок 2.1 – Приклад двоетапної структури виявлення об'єктів: Більш швидкий R-CNN

Основою моделі є згортова нейронна мережа (наприклад, VGG-16 або ResNet), яка створює карти ознак (feature maps). Далі мережа регіональних пропозицій (Region Proposal Network, RPN) формує пропозиції щодо місць, де, ймовірно, знаходяться об'єкти. Потім використовується RoI-пулінг (Region of Interest pooling), який на основі карт ознак та регіональних пропозицій виконує операцію максимального пулінгу, щоб отримати ознаки фіксованого розміру. Ці ознаки подаються на вхід класифікатора, який визначає клас об'єкта та будує його рамку.

Двохетапні детектори зазвичай забезпечують вищу точність порівняно з ранніми одноетапними моделями, проте їхня швидкість роботи недостатня для виконання виявлення об'єктів у реальному часі.

Одноетапні детектори об'єктів пропускають фазу генерації регіональних пропозицій, яка є у двоетапних детекторів, і передбачають

координати обмежувальних рамок та клас об'єкта одночасно. Без значної втрати точності виявлення, одноетапні детектори є досить швидкими для виконання задач реального часу. Два популярних одноетапних детектори – You Only Look Once (YOLO) [16] та Single Shot Detector (SSD) [17] – використовують багатомасштабні карти ознак з Feature Pyramid Networks (FPN) та якірні рамки (anchor boxes) для передбачення класів об'єктів і регресії обмежувальних рамок на сітці клітинок.

З подальшими покращеннями YOLO, точність одноетапних детекторів була значно підвищена, і вони стали популярними завдяки хорошему балансу між точністю та швидкістю роботи. На рисунку 2.2 показано класичний одноетапний детектор YOLOv3. Замість генерації пропозицій місцезнаходжень об'єктів за допомогою нейронних мереж, YOLOv3 ділить зображення на сітку клітинок трьох масштабів, і кожна клітинка передбачає якірні рамки та ймовірності класів.

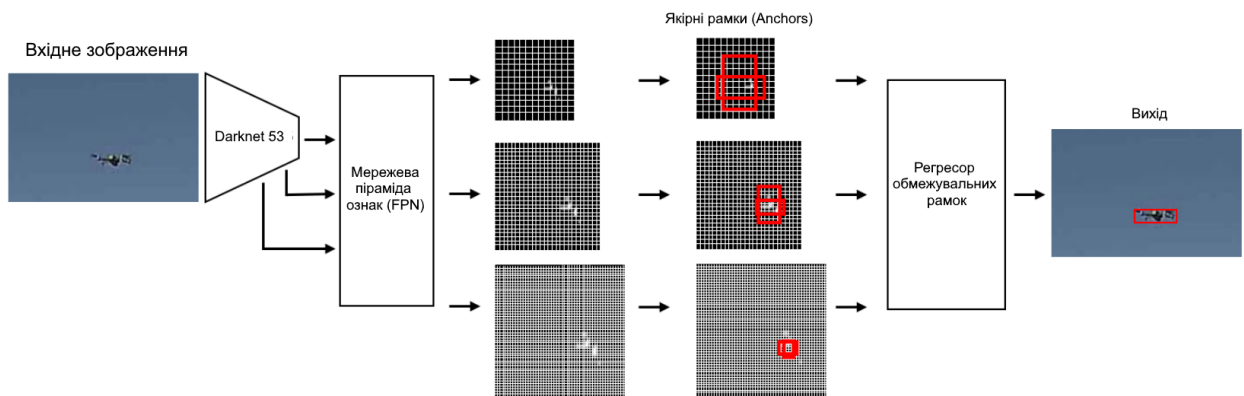


Рисунок 2.2 – Приклад одноетапної структури виявлення об'єктів YOLOv3

За останні кілька років алгоритми серії YOLO були оновлені до версії YOLOv5. Перші три версії YOLO були запропоновані одним і тим самим автором, при цьому YOLOv3 стала значним кроком вперед завдяки суттєвому покращенню точності виявлення та швидкості роботи.

YOLOv3 використовує Darknet53 як основу (backbone) та застосовує Feature Pyramid Networks (FPN) як “шийку” (neck) моделі. Основна мережа складається зі згорткових шарів, які витягують ознаки з вхідного зображення, а “шийка” об’єднує ознаки з попередніх згорткових шарів за допомогою апсемплінгу (збільшення розміру).

Вихідна частина моделі (head) також має пірамідальну структуру, яка формує передбачення обмежувальних рамок (bounding boxes) на трьох різних рівнях: на 82-му, 94-му та 106-му шарах мережі. Відповідно, ці три шари створюють карти ознак розміром 13×13 , 26×26 та 52×52 . Шар 13×13 відповідає за виявлення великих об’єктів, тоді як 52×52 — за виявлення дрібних об’єктів.

На кожній клітинці карти ознак використовується по 3 заздалегідь визначені прив’язки (anchor boxes), центровані в кожній клітинці, для прогнозування рамок. Крім того, у YOLOv3 застосовуються skip-з’єднання для об’єднання залишкових зв’язків між шарами, що покращує точність виявлення.

Класи об’єктів і рівень впевненості у виявленні прогнозуються за допомогою логістичної регресії.

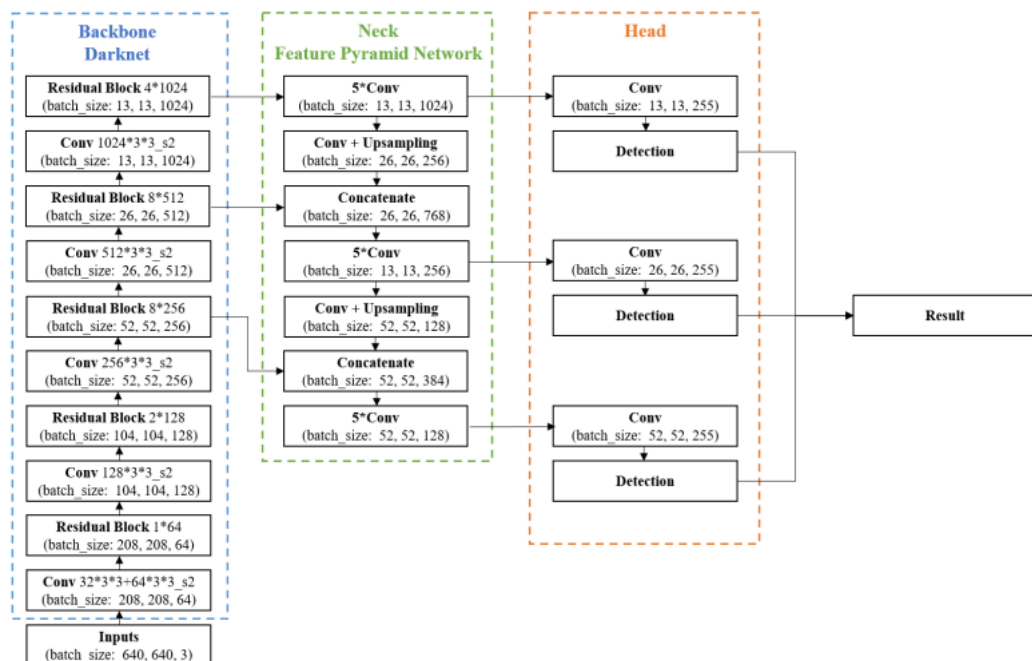


Рисунок 2.3 – Структура YOLOv3

У 2020 році з'явилися дві нові версії YOLO – YOLOv4 та YOLOv5, розроблені різними дослідницькими командами. Обидві моделі показали покращення у продуктивності.

YOLOv4 змінила базову мережу (backbone) на CSPDarknet53 для ефективнішого вилучення ознак, замінила «шийку» моделі YOLOv3 на комбінацію Feature Pyramid Network (FPN) та Path Aggregation Network (PAN), а також досліджувала інші аспекти YOLOv3, включаючи функції активації, функцію втрат для обмежувальних рамок, методи аугментації даних, регуляризації тощо.

YOLOv5 здобула ще більшу популярність, оскільки, окрім змін у базових компонентах мережі, модель була реалізована на Python та PyTorch, на відміну від попередніх версій, написаних на C. Крім того, YOLOv5 пропонує моделі п'яти різних масштабів: N (tiny), S (small), M (medium), L (large) та X (extra-large). Більші моделі мають глибші мережі та ширші канали, тобто містять більше шарів і фільтрів, хоча структура всіх п'яти варіантів загалом залишається однаковою.

2.3 Виявлення малих об'єктів

Однією з основних проблем у задачах виявлення об'єктів є виявлення малих об'єктів. Під малими об'єктами розуміють як ті, що мають невеликі фізичні розміри у реальному світі, так і ті, що виглядають малими на зображенні. У випадку з виявленням і відстеженням дронів проблема відноситься саме до другого типу. Малим вважається об'єкт, розмір якого менше 32×32 пікселів або його висота й ширина займають менше 10% відповідних розмірів зображення.

Малі об'єкти на зображеннях з низькою роздільною здатністю часто не мають чітких контурів чи текстур, що значно ускладнює їх виявлення. Упродовж останніх років було запропоновано багато підходів для покращення виявлення саме малих об'єктів.

Починаючи з YOLOv3, моделі сімейства YOLO використовують структуру під назвою піраміда прогнозування (prediction pyramid), яка об'єднує ознаки, отримані як з глибоких, так і з поверхневих шарів мережі. Це дозволяє робити передбачення на трьох рівнях ієрархічних карт ознак. Глибокі шари мають великі рецептивні поля і багаті на семантичну інформацію, що корисно для виявлення великих об'єктів. Натомість шари з меншим рецептивним полем, розташовані ближче до входу, зберігають точну інформацію про розташування та краї об'єктів, що робить їх придатнішими для виявлення малих об'єктів.

Мережі для виявлення об'єктів мають різні розміри вхідних зображень. Наприклад, для YOLOv3 та YOLOv4 типовий розмір входу становить 416×416 пікселів, а для YOLOv5 – 640×640 пікселів. Якщо зображення має більший розмір, його потрібно масштабувати до відповідного розміру мережі. Наприклад, зображення розміром 1280×1280 пікселів, що містить дрон розміром 16×16 пікселів, після масштабування до 640×640 пікселів зменшить дрон до 8×8 пікселів. Після обробки згортковими шарами та шарами максимум-пулінгу ці ознаки можуть ще більше зменшитись або навіть зникнути, що значно ускладнює виявлення об'єкта.

Один із методів покращення точності для малих об'єктів – це розбиття зображення на фрагменти (tiling). У цьому підході зображення поділяється на менші ділянки (патчі), і кожна з них обробляється мережею окремо [24]. Таким чином можна уникнути масштабування всього зображення, зберігаючи детальні ознаки, оскільки роздільна здатність об'єктів не змінюється. Однак цей метод значно збільшує час обробки, оскільки потрібно обробити кожен фрагмент окремо, а також витратити додатковий час на обрізання зображення та об'єднання передбачених рамок.

3 ВИЯВЛЕННЯ ДРОНІВ ЗА ДОПОМОГОЮ ГЛИБОКОГО НАВЧАННЯ

3.1 Огляд вхідних даних

Дані, використані в цій роботі, складаються з публічно доступних джерел. Публічно доступні дані були завантажені з Інтернету. Усі мітки були створені або перетворені у формат YOLO.

У цій кваліфікаційній роботі були використані два датасети з відкритих онлайн-ресурсів: SimUAV та Wild Bird dataset.

Під час пошуку публічних датасетів для виявлення дронів було встановлено, що лише незначна частина з них містить зображення дронів у малому масштабі або чітко позначає різні типи дронів як окремі категорії. SimUAV – це симульований датасет дронів, створений за допомогою симулятора Airsim та ігрового рушія UE4. Airsim використовується для моделювання польоту дронів, а UE4 – для створення віртуального середовища. У цьому датасеті автор застосував чотири моделі дронів: Parrot A.R. Drone, DJI Inspire I, DJI Mavic 2 Pro та DJI Phantom 4 Pro, і кожна з них була промаркована окремою міткою.

У рамках цієї роботи було використано 4 095 зображень, які містили лише дрони моделі DJI Mavic 2 Pro. Це дозволило дослідити, як змінюється ефективність моделі, якщо для навчання використовуються зображення, зняті іншою камерою, ніж ті, що використані для тестування. Усі зображення в цьому датасеті мають розмір 640×640 пікселів.

Wild Bird dataset був зібраний на вітровій електростанції в Японії за допомогою цифрової фотокамери. Роздільна здатність зображень становить 5616×3744 пікселів. У цьому датасеті міститься 2371 зображення з анотаціями птахів. Ці зображення були використані для дослідження, чи може додавання зображень птахів до навчального набору покращити здатність моделі розрізняти дронів і птахів.

Також у кваліфікаційній роботі було створено реалістичний датасет дронів. Відео з польотами дрона DJI Mavic 2 Pro на відстанях від 5 до 200 метрів були зняті за допомогою камер з роздільною здатністю 1920×1080 пікселів та iPhone (1920×1080 пікселів і 4096×2160 пікселів). Із відео кадри були витягнуті з частотою 15 кадрів на секунду.

Усього було зібрано 9 000 зображень (по 3 000 з кожної камери), причому на кожному зображенні був тільки один дрон. Після впровадження системи з використанням моделі, навченої на цих зображеннях, було виявлено, що іноді модель помилково розпізнавала птахів як дронів.

3.2 Розбиття зображень на фрагменти

Метод розбиття зображень на фрагменти (tiling) полягає в поділі великого зображення на менші частини (патчі), які потім окремо передаються до нейронної мережі для обробки. Такий підхід дозволяє уникнути масштабування всього зображення до фіксованого розміру, що часто призводить до втрати деталей – особливо у випадку з малими об'єктами, такими як дрони.

Завдяки розбиттю зображення на частини, роздільна здатність об'єктів залишається незмінною, і дрібні об'єкти, які могли б зникнути в результаті згорткових операцій або пулінгу, зберігаються з достатнім рівнем деталізації. Це підвищує ймовірність правильного виявлення.

Однак цей метод має й недоліки. Оскільки кожен фрагмент зображення потрібно обробити окремо, час інференсу значно зростає. Крім того, потрібно додатково обчислювати, як обрізати зображення на фрагменти, а після обробки - об'єднувати передбачені рамки назад у загальне зображення. Незважаючи на це, tiling залишається ефективним методом для покращення точності виявлення малих об'єктів у зображеннях з високою роздільною здатністю.

Для підвищення ефективності моделі у виявленні дронів, особливо коли дрон має малий розмір на зображенні, було застосовано метод розбиття зображень на фрагменти (tiling). Щоб уникнути втрати об'єктів, які можуть розміщуватись на краях фрагментів, обрізані частини зображення повинні частково перекриватися. Довжина перекриття має бути більшою за ширину і висоту обмежувальних рамок, що позначені в датасеті.

Роздільна здатність камери становила 1920×1080 пікселів, а вхідний розмір зображення для моделі YOLOv5 – 640×640 пікселів. У межах цієї роботи кожне зображення датасету розбивалося на 8 фрагментів: 2 у вертикальному напрямку та 4 у горизонтальному. Ширина та висота кожного фрагмента становили 640 пікселів, що відповідало розміру входу YOLOv5.

Формула для обчислення розміру перекриття між рівномірно розміщеними фрагментами:

$$L_o = \frac{nL_{\text{tile}} - L_{\text{image}}}{n - 1}. \quad (3.1)$$

де L_o – це довжина перекриття, n – кількість фрагментів у відповідному напрямку, L_{tile} – довжина одного фрагмента (тайлу), а L_{image} – довжина вихідного зображення в тому ж напрямку.

Згідно з цією формулою, довжина перекриття між суміжними фрагментами у вертикальному напрямку становить 200 пікселів, що зручно, оскільки це ціле число. Проте у горизонтальному напрямку формула дає результат $213,3\bar{3}$ пікселя, а така точність є складною для реалізації на практиці.

Замість цього було прийнято рішення вручну задати довжину трьох горизонтальних перекриттів як 200, 200 та 240 пікселів, щоб зробити розбиття зображення більш точним і простішим для обчислення.

Рисунок 3.1 ілюструє, як саме зображення ділиться на фрагменти, а рисунок 3.2 показує, як виглядають окремі фрагменти одного зображення.

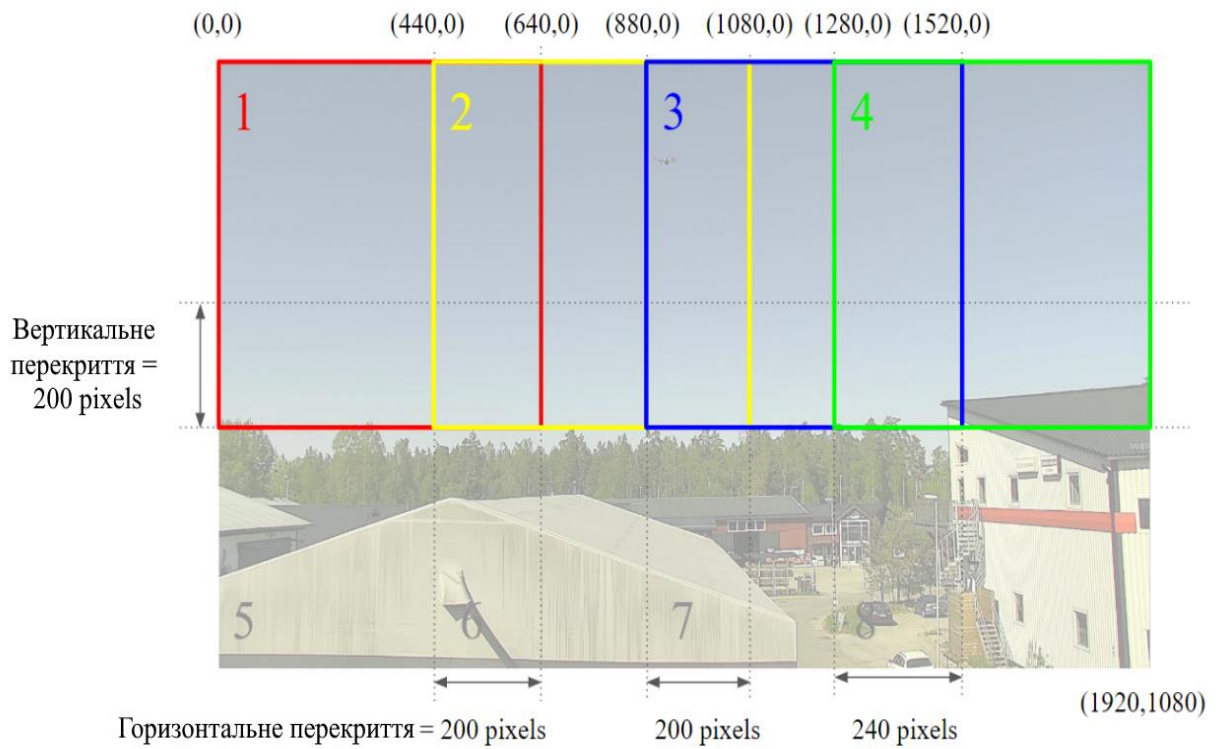


Рисунок 3.1 – Стратегія розбиття зображень на фрагменти

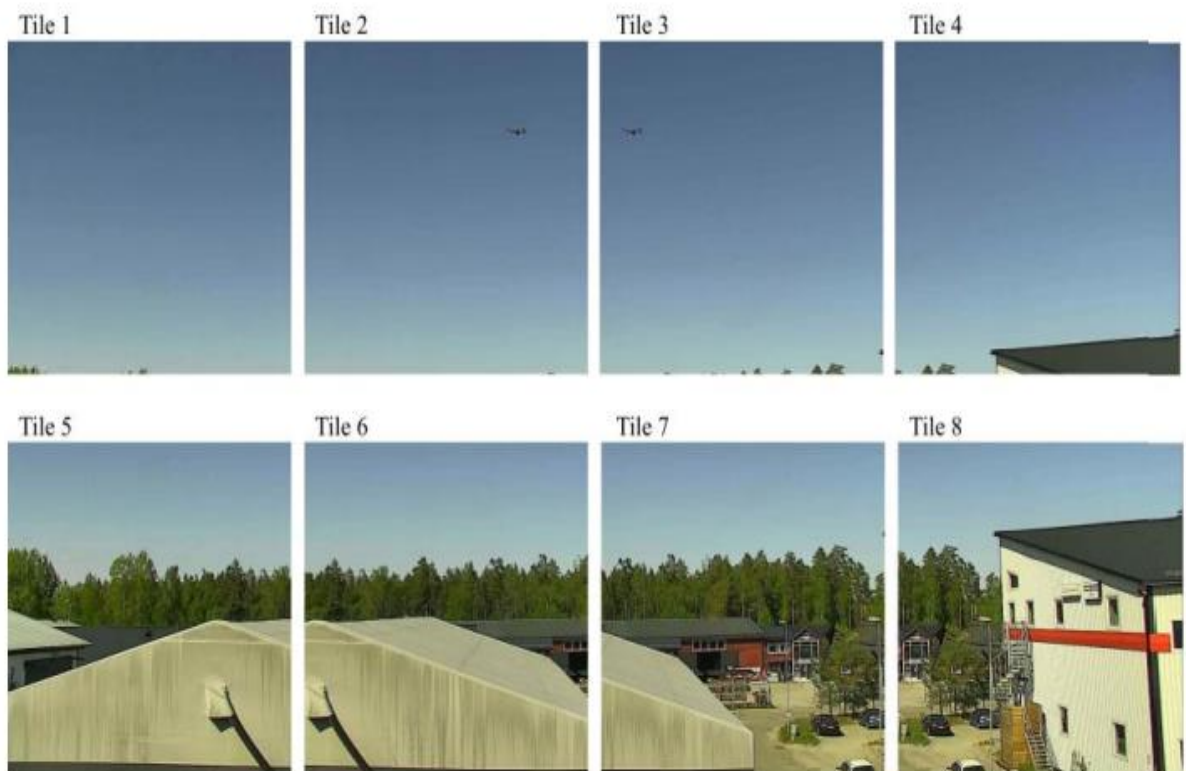


Рисунок 3.2 – Вісім фрагментів одного еталонного зображення

Останні чотири значення – x , y , w , h – обчислюються за формулами, наведеними нижче. У системі координат використовуються піксельні індекси, де один піксель є базовою одиницею. Початок координат знаходиться у верхньому лівому куті зображення, а значення індексів зростають зліва направо та згори донизу.

Клас об'єкта задається як ціле число, а координати та розміри (x , y , w , h) - як дійсні числа від 0 до 1, що обчислюються за такими формулами:

- x_{\min} , y_{\min} – координати верхнього лівого кута обмежувальної рамки;
- x_{\max} , y_{\max} – координати нижнього правого кута рамки;
- w_{image} , h_{image} – ширина та висота зображення відповідно.

Формули розрахунку виглядають наступним чином:

$$x = \frac{x_{\min} + x_{\max}}{2} \cdot \frac{1}{w_{\text{image}}}, \quad (3.2)$$

$$y = \frac{y_{\min} + y_{\max}}{2} \cdot \frac{1}{h_{\text{image}}}, \quad (3.3)$$

$$w = \frac{x_{\max} - x_{\min}}{w_{\text{image}}}, \quad (3.4)$$

$$h = \frac{y_{\max} - y_{\min}}{h_{\text{image}}}. \quad (3.5)$$

Приклад розрахунків можна побачити на рисунку 3.3.

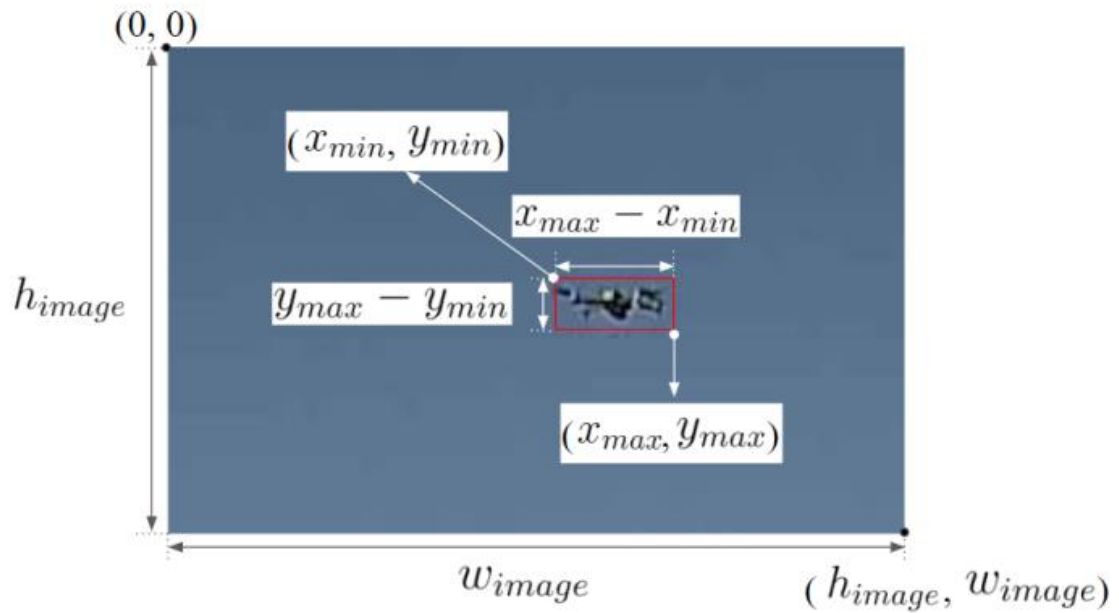


Рисунок 3.3 – Приклад розрахунку для формату YOLO

Оскільки координати x і y , а також висота та ширина зображення змінюються у новому фрагменті (тайлі), необхідно змістити анотації відповідно до того, в якому фрагменті знаходиться об'єкт.

Тобто, під час розбиття зображення на кілька частин, координати об'єкта в новому фрагменті повинні бути перераховані з урахуванням:

- зсуву тайла відносно початкового зображення (тобто зсуву по x та y);
- нових розмірів фрагмента (як правило, фіксованих, наприклад, 640×640 пікселів);
- та обмеження лише на ті об'єкти, які повністю або частково потрапили в межі даного фрагмента.

Цей процес необхідний для збереження коректності YOLO-міток у кожному тайлі, оскільки моделі працюють із нормалізованими координатами об'єктів відносно конкретного зображення, на яке подається тайл.

3.3 Метрики оцінювання

У цій роботі для оцінки продуктивності моделей були використані три метрики: час інференсу, точність (precision) та повнота (recall).

Система відстеження дронів повинна працювати в режимі реального часу, тому середній час інференсу моделей був розрахований за допомогою тестового набору даних. Щоб забезпечити хороший користувацький досвід, кількість кадрів за секунду повинна перевищувати 10, тобто час інференсу моделі має бути меншим за 100 мс на кадр. Крім того, розбиття зображення на фрагменти збільшує кількість запусків моделі. Для кожного зображення в модель подається 8 фрагментів, тому щоб забезпечити достатню швидкість виявлення об'єктів для реального часу, інференс моделі повинен бути меншим за 12,5 мс на фрагмент.

Була застосована метрика Intersection over Union (IoU), щоб виміряти ефективність моделі виявлення об'єктів. IoU обчислюється як площа перетину, поділена на площу об'єднання двох обмежувальних рамок: реальної (ground truth) та передбаченої (predicted) рамки (рисунок 3.4). Реальні обмежувальні рамки анотовані вручну, а передбачені — це вихідні дані моделі.

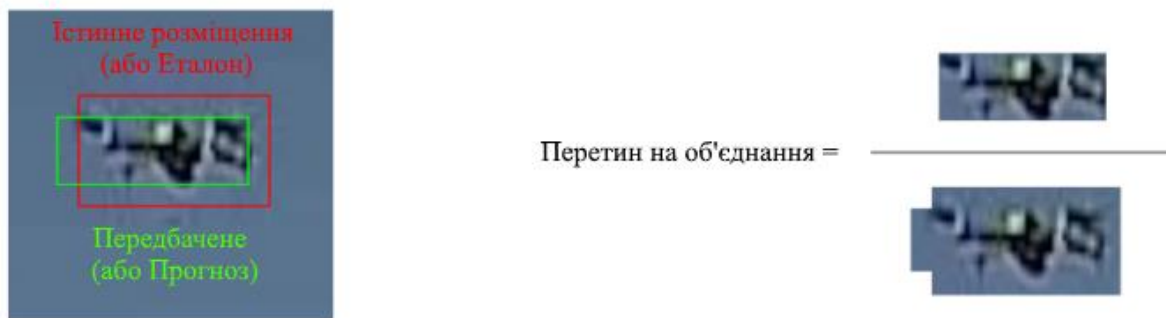


Рисунок 3.4 – Приклад для розрахунку IoU

Чим вищий показник IoU, тим більше передбачена обмежувальна рамка збігається з реальною. Встановлюється порогове значення для визначення, чи є виявлений об'єкт дійсним. У цій роботі поріг встановлено на рівні 0.65. У такому випадку:

- якщо $IoU \geq 0.65$, виявлений об'єкт вважається дійсним і класифікується як істинно позитивний (true positive, TP);
- якщо $IoU \leq 0.65$, виявлений об'єкт вважається недійсним і класифікується як хибно позитивний (false positive, FP) ;
- якщо модель не змогла виявити реальний об'єкт на зображенні, це класифікується як хибно негативний (false negative, FN) ;
- усі частини зображення, де немає ні об'єкта, ні його передбачення, класифікуються як істинно негативні (true negative, TN).

Точність (Precision, P) і повнота (Recall, R) обчислюються на основі кількостей TP, FP та FN.

$$P = \frac{TP}{TP + FP}. \quad (3.6)$$

$$R = \frac{TP}{TP + FN}. \quad (3.7)$$

Точність показує, наскільки правильними є виявлення, а повнота – наскільки повно модель здатна виявити всі об'єкти.

В ідеалі, прагнемо до високих значень як точності, так і повноти, щоб система відстеження дронів не відстежувала неправильні об'єкти і не пропускала дрони. У деяких випадках клієнти можуть не бажати жодних хибних спрацювань і віддаватимуть перевагу моделі з дуже високою точністю, не зважаючи на повноту. Інші клієнти, які хочуть відстежити кожну підозрілу ціль, можуть обрати модель з високою повнотою і допустимим рівнем хибних спрацювань.

3.4 Статистичний аналіз

Було проведено тест Фрідмана та пост-хок тест Немейї для порівняння продуктивності різних моделей у Python. Кожен датасет був перемішаний тричі та умовно позначений як D1, D2 і D3. Наприклад, D1, D2 і D3 були розділені на навчальні, валідаційні та тестові підмножини. Кожна модель навчалась і тестувалась окремо на D1, D2 і D3, а середні значення та стандартні відхилення метрик оцінювання були подані в цьому звіті. Результати кожної метрики для однієї моделі вважалися окремою групою.

Після цього було проведено тест Фрідмана для оцінювання метрик різних моделей на кожному датасеті. Тест Фрідмана використовується для перевірки наявності статистично значущих відмінностей між різними моделями. Він ґрунтується на такій гіпотезі:

- нульова гіпотеза (H_0): Середні значення метрики оцінювання для всіх груп; однакові;
- альтернативна гіпотеза (H_a): Принаймні одне з середніх значень відрізняється від інших.

Якщо р-значення тесту Фрідмана було більшим за 0,05, нульову гіпотезу відхиляли. У такому випадку виконували пост-хок тест Немейї для визначення, між якими саме парами груп є відмінності. Значення р у цьому тесті показують, чи є статистично значуща різниця між двома групами. Якщо р-значення для порівнюваних груп було меншим за 0,05, це свідчило про наявність статистично значущої різниці в оцінюваній метриці між двома моделями.

4 ОПИС ЕКСПЕРЕМЕНТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Опис експерименту

Було проведено два експерименти з метою визначити, яка модель YOLO найкраще підходить для виявлення дронів у режимі реального часу. Перший експеримент був спрямований на порівняння ефективності моделей YOLOv3, YOLOv4 та YOLOv5. Другий експеримент мав на меті перевірити, чи покращиться продуктивність моделей при додаванні або видаленні голови детекції (detection head).

3 000 реальних зображень, зроблених камерою, були випадковим чином розділені на навчальний, валідаційний і тестовий набори даних у пропорції 8 : 1 : 1.

YOLOv5 пропонує моделі п'яти масштабів: N, S, M, L та X, що означає tiny (дуже мала), small (мала), medium (середня), large (велика) та extra-large (дуже велика). Усі п'ять моделей, а також YOLOv3 та YOLOv4, були навчені з нуля протягом 300 епох на GPU з розміром пакета (batch size) 16.

Подальші модифікації були застосовані до моделі, яка показала найкращий результат у першому експерименті. Починаючи з YOLOv3, моделі сімейства YOLO використовують систему прогнозної піраміди (prediction pyramid) як голову мережі для формування обмежувальних рамок. Ця система включає три голови детекції, що обробляють карти ознак на трьох різних масштабах.

У цьому експерименті була додана ще одна голова детекції на нижньому рівні мережі, щоб перевірити, чи покращить це точність виявлення малих об'єктів. Також було видалено голову детекції для великих об'єктів, щоб оцінити, чи вплине це на продуктивність виявлення малих об'єктів.

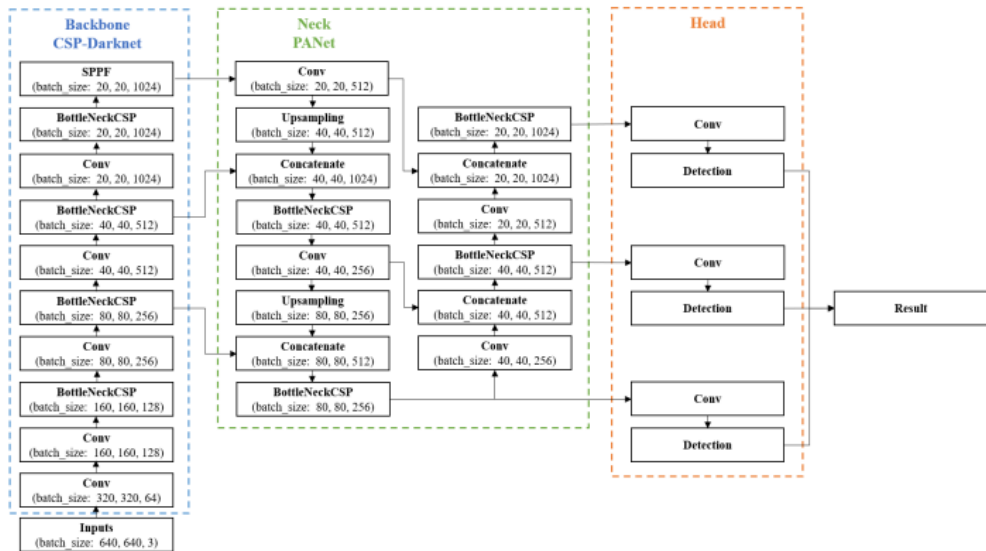


Рисунок 4.1 – Структура YOLOv5

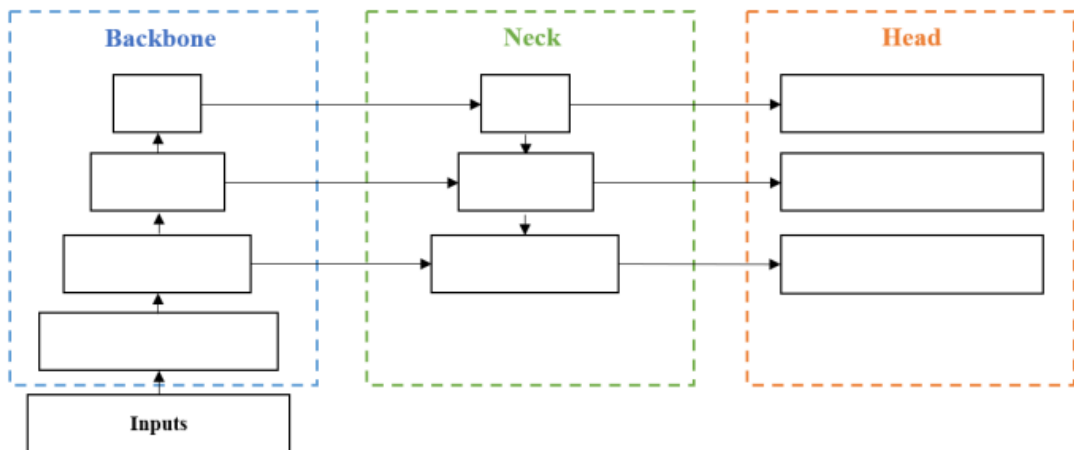


Рисунок 4.2 – Спрощена структура YOLO

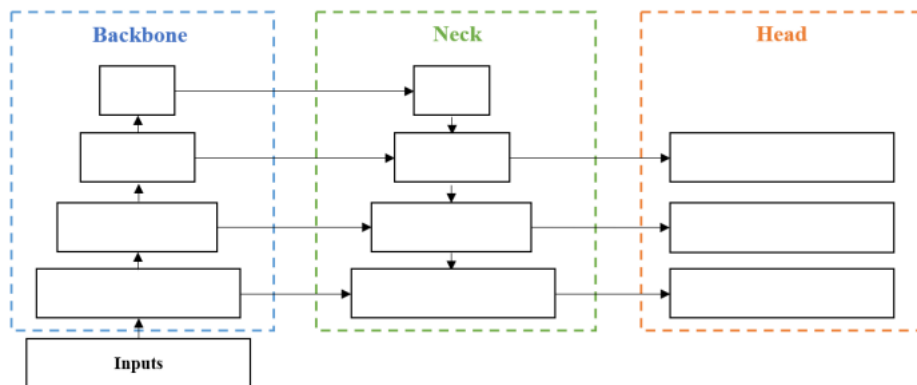


Рисунок 4.3 – Спрощена структура YOLO з додатковою головою детекції

Кожне зображення з навчального, валідаційного та тестового наборів даних у попередніх експериментах було розбите на 8 частин, і мітки дронів були перераховані для кожного фрагмента. Після розбиття було отримано 24 000 зображень. 4 247 зображень містили дрон, тоді як решта були фоновими.

Експеримент порівнював ефективність моделей, навчених на оригінальному датасеті та на розбитому (tiled) датасеті. Також було перевірено вплив навчання без використання фонових зображень.

У цьому експерименті було перевірено, як змінюється продуктивність моделі, якщо вона була навчена на датасеті, знятому іншою камерою, або на датасеті, змодельованому за допомогою комп'ютера.

Далі для експерименту використовувалися зображення, зроблені iPhone у HD-режимі та 4K-режимі, а також симульований датасет SimUAV.

Щоб з'ясувати, скільки нових зображень потрібно, коли модель впроваджується на новій камері, було проведено додатковий експеримент з використанням об'єднаного датасету, що включав зображення, зняті на телефон у HD-режимі, та зображення з камери.

Під час тестування системи відстеження з використанням моделей YOLO було виявлено, що система іноді відстежує птахів. Щоб зменшити кількість хибнопозитивних спрацювань, спричинених птахами, було використано датасет птахів, який використовувався для навчання разом із зображеннями дронів.

Модель була навчена з додатковими зображеннями птахів, які були промарковані як об'єкти переднього плану або фон. Далі з навчального та валідаційного наборів були виключені зображення з надзвичайно малими дронами, щоб перевірити, чи зменшиться кількість хибних спрацювань. Також була протестована модель YOLOv5 з додатковими головами, щоб перевірити, чи працює ця архітектура ефективно без наявності надто малих цільових об'єктів.

Моделі були розроблені та протестовані в середовищах, наведених у таблиці 4.1.

Таблиця 4.1 – Технічні характеристики навколишнього середовища

| Назва | Версія |
|-------------|--------|
| Ubuntu | 18.04 |
| CUDA | 10.2 |
| OpenCV | 4.1.1 |
| torch | 1.8.0 |
| torchvision | 0.9.0 |
| numpy | 1.13 |
| matplotlib | 3.3.4 |
| pandas | 0.22.0 |
| scipy | 0.19.1 |

Всі моделі YOLO в цій роботі використовують однакові налаштування гіперпараметрів.

Таблиця 4.2 – Налаштування гіперпараметрів

| Гіперпараметри | Знач. | Коментарі |
|-----------------|--------|---|
| 1 | 2 | 3 |
| lr0 | 0.01 | початкова швидкість навчання |
| lrf | 0.01 | фінальна швидкість навчання OneCycleLR ($lr0 \times lrf$) |
| momentum | 0.937 | імпульс SGD / параметр β_1 в Adam |
| weight_decay | 0.0005 | згасання ваг оптимізатора |
| warmup_epochs | 3.0 | кількість епох розігріву |
| warmup_momentum | 0.8 | імпульс при розігріві |
| warmup_bias_lr | 0.1 | початкова швидкість навчання для bias |
| box | 0.05 | коефіцієнт втрат по рамці (box loss gain) |
| cls | 0.5 | коефіцієнт втрат по класу (classification loss gain) |

Продовження таблиці 4.2

| 1 | 2 | 3 |
|-------------|-------|--|
| cls_pw | 1.0 | ваговий коефіцієнт для ВСЕ втрат по класу |
| obj | 1.0 | коефіцієнт втрат по об'єкту (масштабується пікселями) |
| obj_pw | 1.0 | ваговий коефіцієнт для ВСЕ втрат по об'єкту |
| iou_t | 0.20 | поріг для IoU при навчанні |
| anchor_t | 4.0 | поріг для multiple-anchor |
| anchors | 3 | якірів на кожен вихідний шар (0 – ігнорувати) |
| fl_gamma | 0.0 | гамма-фактор для фокальної втрати (default efficientDet = 1.5) |
| hsv_h | 0.015 | зміна тону (Hue) зображення (коефіцієнт) |
| hsv_s | 0.7 | зміна насиченості (Saturation) зображення |
| hsv_v | 0.4 | зміна яскравості (Value) зображення |
| degrees | 0.0 | обертання зображення (+/- градуси) |
| translate | 0.1 | зсув зображення (+/- частка) |
| scale | 0.5 | масштабування зображення (+/- коефіцієнт) |
| shear | 0.0 | зсув (shear) зображення (+/- градуси) |
| perspective | 0.0 | перспектива зображення (коефіцієнт, діапазон 0–0.001) |
| flipud | 0.0 | віддзеркалення по вертикалі (ймовірність) |
| fliplr | 0.5 | віддзеркалення по горизонталі (ймовірність) |
| mosaic | 1.0 | мозаїчне перетворення (ймовірність) |
| mixup | 0.0 | міхур-аугментація (ймовірність) |
| copy_paste | 0.0 | сегментне копіювання-вставлення (ймовірність) |

4.2 Вибір моделі SOTA

Різні результати продуктивності моделей YOLO наведені в таблиці 4.3. Найбільша версія YOLOv5 (YOLOv5x) аварійно завершувала роботу щоразу під час навчання, тому її результати відсутні.

У колекції моделей YOLOv5 спостерігається така тенденція: чим більша модель, тим кращі показники точності (precision) та повноти (recall). Серед усіх отриманих коректних результатів YOLOv5l має найвищу точність і повноту, однак її швидкодія є недостатньою для використання з методом розбиття зображень на фрагменти (tiling).

Модель YOLOv5n є найшвидшою серед усіх і єдиною, яка задовольняє вимоги за часом для використання з методом tiling.

Таблиця 4.3 – Результати тестування моделей YOLO

| Модель | Точність (Precision, %) | Повнота (Recall, %) | Час інференсу (мс) |
|---------|----------------------------|------------------------|--------------------|
| YOLOv3 | 93.1 ± 0.6 | 75.1 ± 0.4 | 102.1 ± 0.2 |
| YOLOv4 | 93.3 ± 0.8 | 74.1 ± 0.6 | 25.0 ± 0.3 |
| YOLOv5n | 93.8 ± 0.5 | 72.6 ± 0.6 | 7.9 ± 0.2 |
| YOLOv5s | 95.4 ± 0.5 | 74.1 ± 0.8 | 19.4 ± 0.3 |
| YOLOv5m | 96.1 ± 0.3 | 75.6 ± 0.6 | 49.5 ± 0.2 |
| YOLOv5l | 98.8 ± 0.3 | 76.4 ± 0.2 | 101.4 ± 0.3 |
| YOLOv5x | N/A | N/A | N/A |

P-значення тестів Фрідмана для точності (precision), повноти (recall) та часу інференсу були меншими за 0.05, що свідчить про наявність статистично значущих відмінностей між моделями за всіма трьома метриками.

Для виявлення пар моделей, які мають статистично різні середні значення точності, повноти та часу інференсу, було проведено тест Немейї. Результати тесту були візуалізовані на рисунку 4.4.

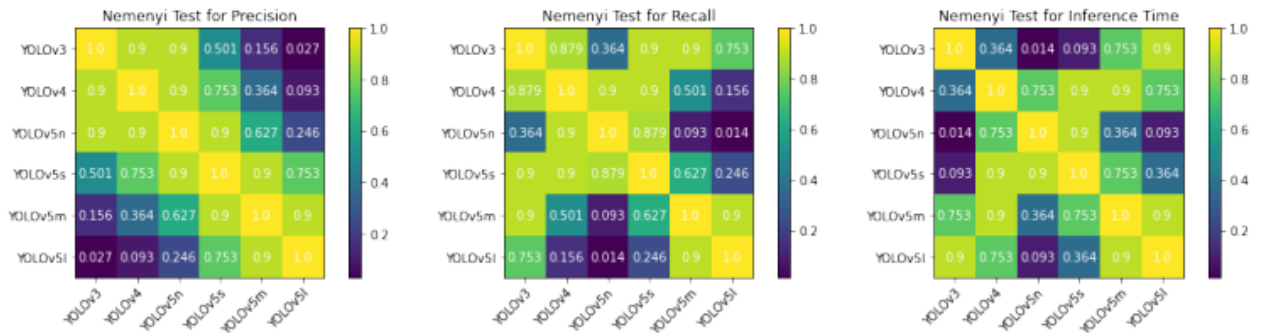


Рисунок 4.4 – Тест Немейї для результатів тестування моделей сімейства YOLO

На рисунку 4.4 зліва лише моделі YOLOv3 та YOLOv5l мали р-значення менше за 0.05, отже, тільки між цими двома моделями була статистично значуща різниця середніх значень точності (precision).

Щодо повноти (recall), моделі YOLOv5l та YOLOv5n мали р-значення $0.014 < 0.05$, тобто повнота YOLOv5l була статистично вищою, ніж у YOLOv5n.

Хоча середній час інференсу YOLOv5n був значно меншим порівняно з іншими моделями, згідно з тестом Немейї він був статистично меншим лише у порівнянні з YOLOv3.

На відміну від очікувань, додавання голови детекції, яка використовувала карти ознак з вищою роздільною здатністю, негативно вплинуло на продуктивність моделі за всіма метриками оцінюванн. Після видалення голови детекції для великих об'єктів швидкість інференсу та повнота дещо покращилися, але все одно залишалися гіршими, ніж у початкової моделі.

Результати наведено в таблиці 4.4.

LOv5n-P2345 – модель із додатковою головою детекції.

YOLOv5n-P234 – модель без голови детекції для великих об'єктів.

Таблиця 4.4 – Результати тестування моделі YOLOv5n та її варіацій

| Модель | Точність (Precision, %) | Повнота (Recall, %) | Час інференсу (мс) |
|---------------|-------------------------|---------------------|--------------------|
| YOLOv5n | 93.8 ± 0.5 | 72.6 ± 0.6 | 7.9 ± 0.2 |
| YOLOv5n-P2345 | 93.4 ± 0.7 | 71.2 ± 0.4 | 10.5 ± 0.3 |
| YOLOv5n-P234 | 90.0 ± 0.8 | 71.0 ± 0.8 | 9.9 ± 0.3 |

P-значення тестів Фрідмана для точності (precision) та часу інференсу були меншими за 0.05, що свідчить про статистично значущі відмінності між моделями за цими метриками. У той же час, повнота (recall) не мала статистично значущої різниці між моделями.

Згідно з результатами пост-хок тесту Немейї, показаними на рисунку 4.5, точність моделі YOLOv5n-P234 була статистично нижчою, ніж у YOLOv5n, а час інференсу моделі YOLOv5n-P2345 був статистично більшим, ніж у YOLOv5n.

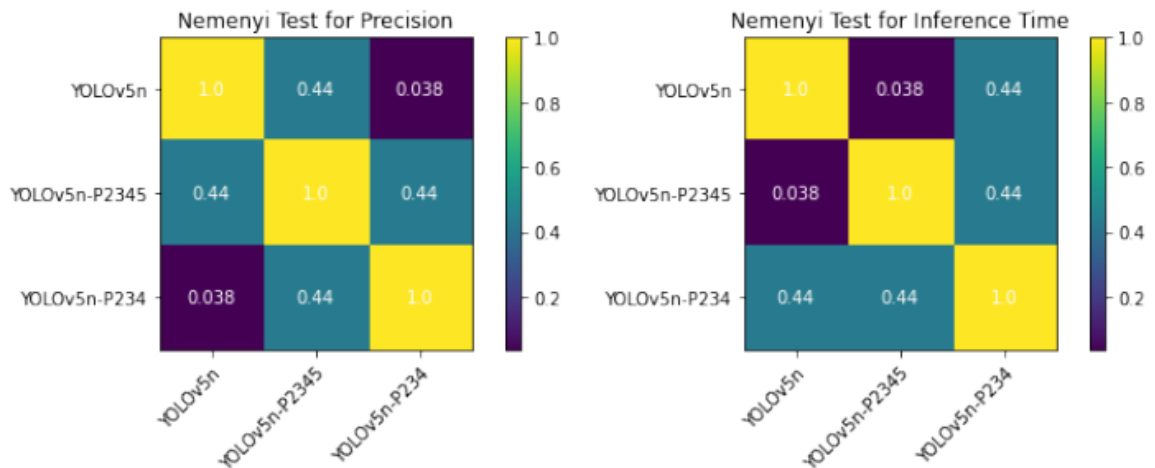


Рисунок 4.5 – Тест Немейї для результатів тестування моделі YOLOv5n та її варіацій

Оригінальний датасет зроблений на камеру складався з 3 000 зображень і 3 000 міток. Після розбиття на фрагменти (tiling) кількість

зображень у датасеті зростає в 8 разів. Кількість анотацій також збільшилася, оскільки деякі дрони потрапляли в області перекриття між фрагментами. У результаті датасет tiled містить 24 000 зображень. З них 19 726 зображень були фоновими, а решта 4 274 зображення були промарковані дронами.

Порівнюючи розташування міток на рисунках 4.6 і 4.7, можна побачити, що дрони почали з'являтися ближче до кутів зображень, а не лише по центру, як у початковому датасеті. Таким чином, після розбиття дрони розподілилися більш рівномірно по всьому зображенню.

Крім того, масштаб координат і відносний розмір об'єктів став більшим на правому зображенні рисунка 4.7, що означає, що той самий об'єкт займає більшу частину зображення після tiling. Отже, відносні розміри об'єктів стали більшими.

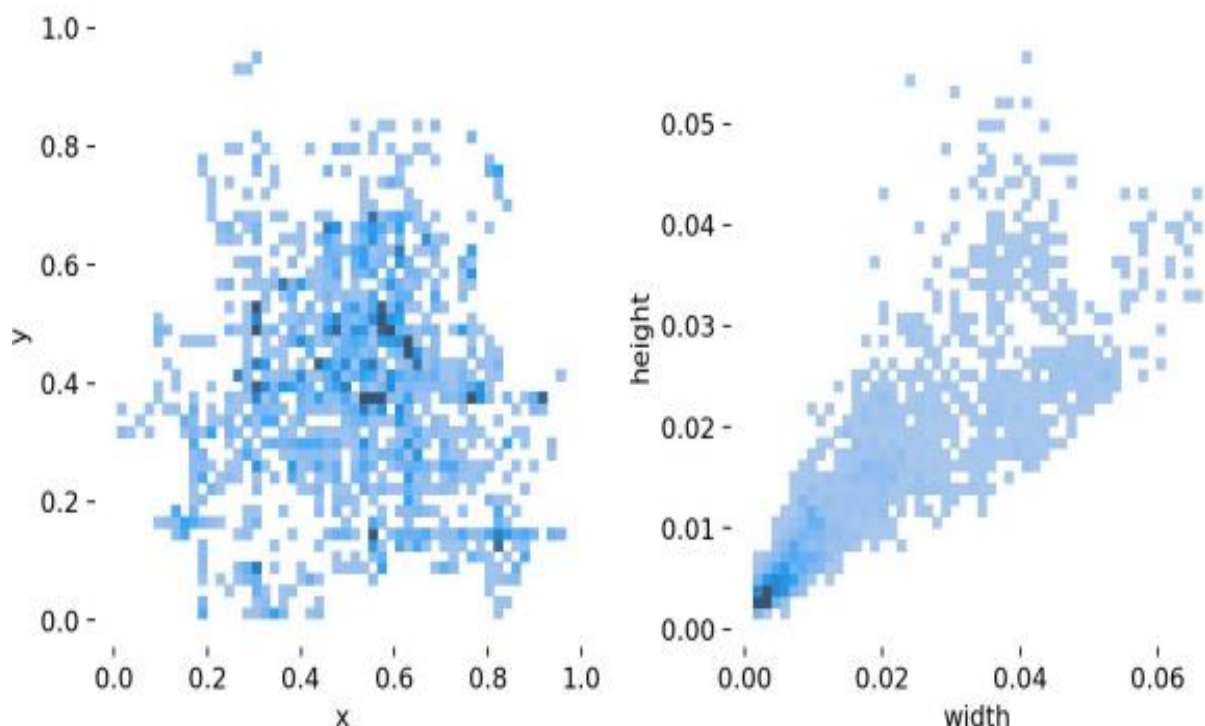


Рисунок 4.6 – Візуалізація даних датасету зробленого на камеру без розбиття на фрагменти: розташування міток (зліва) та відносний розмір об'єктів (справа)

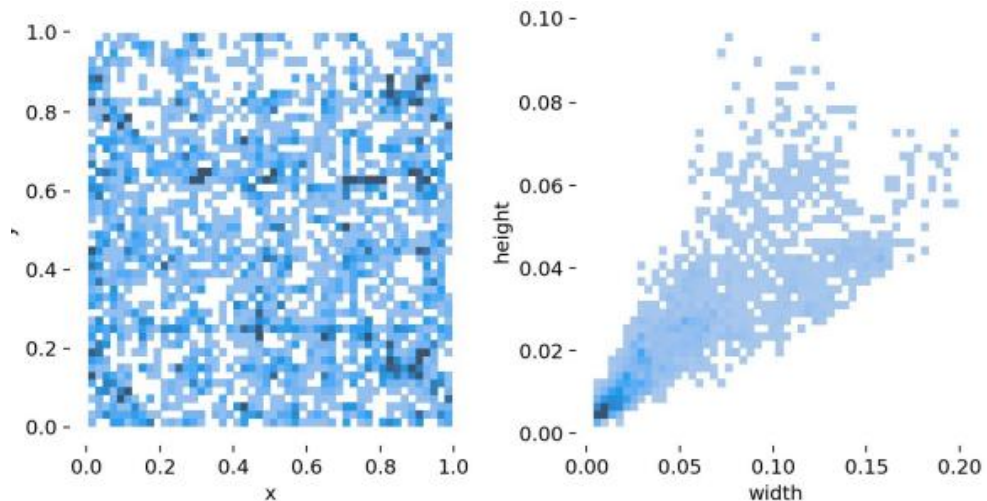


Рисунок 4.7 – Візуалізація даних датасету зробленого на камеру після розбиття на фрагменти (tiling): розташування міток (зліва) та відносний розмір об’єктів (справа)

Модель YOLOv5n була навчена та протестована на оригінальних і розбитих (tiled) зображеннях, з фоновими зображеннями та без них. Результати експериментів наведено в таблиці 4.6 нижче.

Таблиця 4.5 – Скорочення для навчального, валідаційного та тестового наборів

| Скорочення | Навчання (Train) | Валідація (Val) | Тестування (Test) |
|------------|-----------------------------|-----------------------------|------------------------|
| ООО | Оригінальні зображення | Оригінальні зображення | Оригінальні зображення |
| ООТ | Оригінальні зображення | Оригінальні зображення | Розбиті зображення |
| ТТТ | Розбиті зображення | Розбиті зображення | Розбиті зображення |
| ttT | Розбиті зображення без фону | Розбиті зображення без фону | Розбиті зображення |

Таблиця 4.6 – Результати тестування при навчанні на оригінальних або розбитих (tiled) зображеннях

| Скорочення | Точність (Precision, %) | Повнота (Recall, %) |
|------------|-------------------------|---------------------|
| ООО | 93.8 ± 0.5 | 72.6 ± 0.6 |
| ООТ | 72.9 ± 0.8 | 72.4 ± 0.6 |
| ТТТ | 93.3 ± 0.2 | 96.0 ± 0.3 |
| ттТ | 92.6 ± 0.3 | 96.4 ± 0.4 |

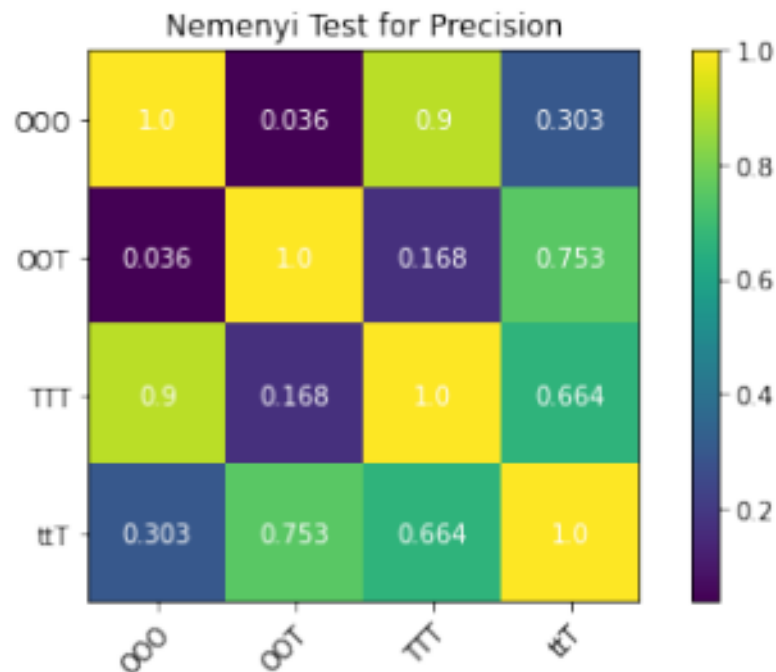


Рисунок 4.8 – Тест Немейї для результатів тестування моделей, навчених на оригінальних або розбитих (tiled) зображеннях

Повнота (recall) дещо зросла, тоді як точність (precision) знизилася, коли модель, навчена на оригінальних зображеннях, була протестована на розбитих (tiled) зображеннях. Це сталося через те, що після розбиття відносний розмір об'єкта збільшився, і тому знання, які модель здобула на оригінальних зображеннях, виявилися не повністю придатними для tiled-

зображень, що призвело до зростання кількості хибнопозитивних виявлень.
Приклад показано на рисунку 4.9.



а) Оригінальне зображення, на якому є дрон по центру.
Модель, навчена на зображеннях оригінального розміру, не змогла виявити дрон.



б) Фрагментоване зображення (тайл) з оригінального зображення.
Модель, навчена на зображеннях оригінального розміру, помилково визначила об'єкт як дрон (false positive).



с) Фрагментоване зображення (тайл) з оригінального зображення.
Модель, навчена на зображеннях оригінального розміру, успішно виявила дрон.

Рисунок 4.9 – Модель, навчена на зображеннях оригінального розміру,
протестована на розбитих зображеннях.

Коли всі розбиті зображення були використані для навчання, повнота значно зросла. Завдяки більшому відносному розміру об'єкта на кожному зображенні, дрон легше розпізнавався моделлю. Проте точність знизилась на 0.01, що свідчить про те, що модель почала частіше помилятися, приймаючи інші об'єкти за дронів.

Продуктивність моделі практично не знизилась, навіть якщо в навчанні не використовувалися 19 726 фонових зображень. Навпаки, повнота навіть зросла, що означає, що модель пропускала менше дронів. Приклад цього наведено на рисунку 4.10. У подальших експериментах фонові зображення будуть вилючені, щоб зекономити місце та час.



а) Оригінальне зображення, на якому дрон розташований на краю даху в центрі. Модель, навчена на зображеннях оригінального розміру, не змогла виявити дрон.



б) Розбите (тайловане) зображення з оригінального. Модель, навчена на фрагментах зображення, виявила хибно-негативний результат (угорі ліворуч), а також помилково виявила дрон (хибно-позитивний результат унизу праворуч).

Рисунок 4.10 – Модель, навчена на розбитих зображеннях, була протестована на розбитих зображеннях

Тести Фрідмана та Немейї також підтвердили цей аналіз. Р-значення тесту Фрідмана для повноти становило $0.060 > 0.05$, отже, нульову гіпотезу про те, що всі моделі мають однакову середню повноту, не можна відхилити.

Щодо точності, лише модель, навчена на оригінальних зображеннях, показала статистично значущу різницю між результатами на тестовому наборі оригінальних і розбитих зображень.

ВИСНОВКИ

У даній кваліфікаційній роботі було досліджено ефективність різних моделей YOLO для задачі виявлення дронів у режимі реального часу з використанням методів глибокого навчання. Було проведено серію експериментів із порівняння моделей YOLOv3, YOLOv4 та п'яти масштабних версій YOLOv5 (N, S, M, L, X) з метою визначення оптимальної архітектури з погляду точності, повноти та швидкодії.

Основні висновки:

- YOLOv5n виявилася найпридатнішою моделлю для виявлення дронів у реальному часі завдяки найменшому часу інференсу (7.9 мс) при достатньо високій точності та повноті. Усі великі моделі, хоча й показали кращу точність, виявилися занадто повільними для практичного використання в системі з обмеженим обчислювальним ресурсом;

- модифікації структури YOLOv5n шляхом додавання або видалення голови детекції не призвели до покращення результатів. Навпаки, точність зменшилася, а швидкість погіршилася. Це підтверджено статистичними тестами Фрідмана та Немейї;

- використання методу розбиття зображень (tiling) значно покращує виявлення малих об'єктів, таких як дрони, які перебувають на великій відстані. Повнота виявлення при цьому зростає з 72.6% до 96.4%, що підтверджує ефективність даного підходу;

- модель, навчена на tiled-зображеннях, навіть без фонових зображень, зберігала високу якість виявлення, а в деяких випадках – навіть покращила повноту. Це дозволяє зменшити розмір датасету та час на його обробку, що є важливим у промисловому застосуванні;

- навчання моделі на даних з різних камер і симульованих зображеннях (SimUAV) дозволяє підвищити узагальнюючу здатність моделі,

що є критично важливим для масштабування системи виявлення дронів на різне обладнання;

- додавання зображень птахів до навчального датасету дозволило зменшити кількість хибнопозитивних спрацювань, викликаних плутаниною між птахами та дронами, що покращує надійність системи.

Отже, результати дослідження підтверджують доцільність застосування компактної версії YOLOv5 (YOLOv5n) у поєднанні з методом розбиття зображень для задачі виявлення дронів у реальному часі. Крім того, було показано, що розумне формування датасету (виключення фону, додавання негативних прикладів, використання симуляцій) істотно впливає на ефективність моделі без потреби у зміні самої архітектури. Це дозволяє створити швидку, точну та адаптивну систему виявлення БПЛА для промислового використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Salman, S., Mir, J., Farooq, M. T., Malik, A. N., & Haleemdeen, R. (2021, January). Machine learning inspired efficient audio drone detection using acoustic features. In 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST) (pp. 335-339). IEEE.
2. Sun, Y., Abeywickrama, S., Jayasinghe, L., Yuen, C., Chen, J., & Zhang, M. (2020). Micro-Doppler signature-based detection, classification, and localization of small UAV with long short-term memory neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(8), 6285-6300.
3. Opromolla, R., & Fasano, G. (2021). Visual-based obstacle detection and tracking, and conflict detection for small UAS sense and avoid. *Aerospace Science and Technology*, 119, 107167.
4. Unlu, E., Zenou, E., & Riviere, N. (2018, January). Using shape descriptors for UAV detection. In *Electronic Imaging 2017* (pp. pp-1).
5. Huang, B., Li, J., Chen, J., Wang, G., Zhao, J., & Xu, T. (2023). Anti-UAV410: A thermal infrared benchmark and customized scheme for tracking drones in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5), 2852-2865.
6. Unlu, E., Zenou, E., Riviere, N., & Dupouy, P. E. (2019). Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSJ Transactions on Computer Vision and Applications*, 11, 1-13.
7. S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small uas," in 2016 International Conference on Unmanned Aircraft Systems (ICUAS), 2016. doi: 10.1109/ICUAS.2016.7502513 pp. 1254–1260
8. A. Shoufan, H. M. Al-Angari, M. F. A. Sheikh, and E. Damiani, "Drone pilot identification by classifying radio-control signals," *IEEE Transactions on*

Information Forensics and Security, vol. 13, no. 10, pp. 2439–2447, 2018. doi: 10.1109/TIFS.2018.2819126

9. Wu, Q., & Zhou, Y. (2019, May). Real-time object detection based on unmanned aerial vehicle. In *2019 IEEE 8th data driven control and learning systems conference (DDCLS)* (pp. 574-579). IEEE.

10. Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., & Matson, E. T. (2020). Real-time and accurate drone detection in a video with a static background. *Sensors*, 20(14), 3856.

11. de Nadal, E., & Posas, F. (2022). The HOG pathway and the regulation of osmoadaptive responses in yeast. *FEMS yeast research*, 22(1), foac013.

12. Лізунов, П., Білощицький, А., Кучанський, О., & Андрашко, Ю. (2021). Комбіновані методи ідентифікації неповних дублікатів у наукових публікаціях. *Управління розвитком складних систем*, (48), 85-94.

13. Setta, Showmik, et al. "Real-time facial recognition using SURF-FAST." *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2021, Volume 2*. Singapore: Springer Singapore, 2021. 505-522.

14. Seetharaman, K., and T. Mahendran. "Leaf disease detection in banana plant using gabor extraction and region-based convolution neural network (RCNN)." *Journal of The Institution of Engineers (India): Series A* 103.2 (2022): 501-507.

15. Maity, M., Banerjee, S., & Chaudhuri, S. S. (2021, April). Faster r-cnn and yolo based vehicle detection: A survey. In *2021 5th international conference on computing methodologies and communication (ICCMC)* (pp. 1442-1447). IEEE.

16. Badgujar, C. M., Poullose, A., & Gan, H. (2024). Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223, 109090.

17. Wen, G., Cao, P., Wang, H., Chen, H., Liu, X., Xu, J., & Zaiane, O. (2023). MS-SSD: Multi-scale single shot detector for ship detection in remote sensing images. *Applied Intelligence*, 53(2), 1586-1604.

18. Мартовицький В.О., Шеховцов О.В., Алейник Д.С., Пахомова Є.В. та Іванченко Д.І. «Підхід до виявлення та класифікації радіокерованих моделей за їх радіосигналом» Вісник Херсонського національного технічного університету» для розміщення у № 2 (2025) препринт.