

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет імені Олеся Гончара
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет імені Олеся Гончара

Кваліфікаційна наукова
праця на правах рукопису

СИЗОНЕНКО ОЛЕКСАНДРА ДМИТРІВНА

УДК 004.42

ДИСЕРТАЦІЯ
РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ ТА ПРОГРАМНИХ
ЗАСОБІВ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ
У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

12 Інформаційні технології

121 Інженерія програмного забезпечення

Подається на здобуття ступеня доктора філософії. Дисертація містить результати власних досліджень. Використання ідей, результатів та текстів інших авторів мають посилання на відповідне джерело.

_____ О.Д. Сизоненко

Науковий керівник:
Божуха Лілія Миколаївна
кандидат фізико-математичних наук, доцент

Дніпро – 2024

АНОТАЦІЯ

Сизоненко О.Д. Розроблення технології та програмних засобів виявлення та розпізнавання об'єктів у режимі реального часу – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення – Дніпровський національний університет імені Олеся Гончара, Дніпро, 2024.

Дисертаційна робота присвячена розробленню ефективної та швидкої інформаційної технології та програмного засобу виявлення, розпізнавання та відстеження об'єктів у режимі реального часу.

Дослідження розкриває сутність технологій глибинного навчання та спрямоване на аналіз поточного стану та виявлення методів покращення роботи програмного рішення виявлення, розпізнавання та відстеження об'єктів у режимі реального часу. У вступі визначено мету, об'єкт, предмет та методи дослідження. Розкрито наукову новизну дослідження, теоретичне та практичне значення наукових результатів, особистий внесок, зазначено інформацію про впровадження і апробацію результатів.

Актуальність теми дослідження обумовлено існуючим протиріччям між ускладненням моделей глибинного навчання при використанні існуючих програмних рішень, з одного боку, та зростанням труднощів інтерпретації цих моделей у прикладному застосуванні з підвищенням швидкодії процедури розпізнавання та відстеження об'єктів у режимі реального часу – з іншого боку. Сфера застосування безпілотних літальних апаратів (БпЛА) постійно зростає і виникає необхідність адаптації алгоритмів машинного навчання та моделей глибинного навчання для використання саме на борту БпЛА при врахуванні обмеженості на обчислювальні ресурси та специфіку організації

отриманого відеопотоку даних. У роботі підкреслюється вага вже існуючих архітектурних рішень реалізації згорткової нейронної мережі при використанні методів та алгоритмів машинного навчання. Встановлене протиріччя долається внесенням додаткових умов для задачі опрацювання великого набору зображень відеопотоку, який постійно оновлюється та може вміщувати інформацію про об'єкти різного розміру та форми для задачі розпізнавання та відстеження. Виділяється необхідність використання в запропонованому програмному рішенні адаптованої функції втрат для підтримки прийняття рішень на основі спостережених даних. Актуальною в цій тематиці є задача злиття ознак згорткових нейронних мереж для просторового та часового потоків.

Метою роботи є підвищення точності виявлення, розпізнавання та відстеження об'єктів в режимі реального часу та реалізації відповідної технології у вигляді програмного засобу.

Об'єктом дослідження є процеси опрацювання даних в задачах виявлення, розпізнавання та відстеження об'єктів у режимі реального часу.

Предметом дослідження є моделі, алгоритми та технології використання згорткових нейронних мереж для вирішення задачі розпізнавання та відстеження об'єкта у режимі реального часу.

У першому розділі дисертації акцентується увага на розгляді предметної області та аналізі сучасних методів виявлення, розпізнавання та відстеження об'єктів. Особлива увага приділена етапам розвитку нейронних згорткових мереж та моделям глибокого навчання для проведення аналізу побудованих архітектурних рішень та використаних алгоритмів. Також розглянуто проблеми, пов'язані з реалізацією інтелектуальних систем у режимі реального часу, включаючи оброблення великого обсягу даних, та оптимізацію швидкості роботи програмних засобів.

У другому розділі дослідження увага акцентується на обґрунтуванні вибраних методів виявлення, розпізнавання та відстеження об'єктів.

Приділена увага етапу підготовки наборів даних (препроцесінг) для їх подальшого використання у моделі. Розглядаються алгоритми сімейств YOLO та Faster R-CNN. Описано деталі навчання даних алгоритмів та наведено результати досліджень. Основну увагу приділено питанню еволюції алгоритму YOLO, що є ключовим для розуміння структури даного алгоритму та використаних методів і підходів організації даних.

У третьому розділі основна увага приділена реалізації технології розпізнавання та відстеження об'єктів в режимі реального часу, а також аналізу отриманих результатів роботи розробленої технології при порівнянні з існуючими програмними рішеннями. Проведено огляд існуючих наборів даних для подальшої апробації алгоритмів та обговорюється питання анотування даних та формування власного набору даних (dataset, датасет). Експериментальне дослідження на визначених наборах даних (зокрема, власний, Kitti та TAIVD) з невеликою кількістю зображень показало, що запропонована технологія у вигляді комбінації алгоритмічних рішень з двох типів згорткових нейронних мереж, демонструє покращену приблизно на 1,8 % точність розпізнавання ніж технологія-аналог YOLO. Отримані результати спостережень з побудованого власного набору даних з невеликою кількістю зображень мають перевагу за розміром моделі на 6 % відносно датасету TAIVD в тому ж об'ємі, який може бути доречним для використання за рахунок вміщення зображень з БПЛА. При додаванні до базової моделі шару детектування B2-PFNB при невеликій кількості зображень усереднена точність mAP збільшується на 4,3%. Кожна обрана стратегія покращення технології-аналога YOLO різною мірою покращила ефективність виявлення, розпізнавання та відстеження об'єктів в режимі реального часу.

Внутрішня логіка програмного забезпечення розробленої технології побудована з використанням компонентно-орієнтованого підходу. Для реалізації обчислень використано такі бібліотеки, як Pytorch, Ultralytics,

NumPy, OpenCV2 та Matplotlib. Для розгортання цієї технології запропоновано реалізація декількох точок інтеграції.

Наукова новизна результатів дослідження полягає в наступному:

- *вперше* розроблено архітектурне рішення побудови нейронної згорткової мережі задачі виявлення, розпізнавання та відстеження об'єктів в режимі реального часу, що відрізняється від існуючого рішення тим, що використовує більшу кількість блоків розпізнавання об'єктів різного розміру, яке є оптимізоване для задач конкретної предметної області;

- *вперше* обґрунтовано можливість використання в розробленій технології PFNB-блоку, який базується на архітектурному рішенні Faster-Net, що використовує багатомасштабну мережу об'єднання ознак для демонстрації покращеної точності розпізнавання у порівнянні з базовою технологією;

- *вперше* сформований власний набір даних для апробації розробленої технології починаючи з етапу розпізнавання об'єктів у відеопотоці, який включає об'єкти різного масштабу визначеної предметної області, що підтверджує ефективність розробленої моделі;

- *вперше* запропоновано архітектуру кросплатформної бібліотеки для реалізації технології виявлення, розпізнавання та відстеження об'єктів, яка є п'ятимасштабною структурою і містить механізм уваги ViFormer з малою обчислювальною потужністю, що дозволяє покращити точність виявлення малих об'єктів та покращує увагу до ключової інформації на карті об'єктів;

- *вперше* проведено моделювання порівняльних експериментів на YOLO v9 на невеликому наборі даних, які відрізняються використанням різних видів функцій втрат при зберіганні інших умов навчання незмінними, що показало використання функції регресійних втрат WIoU v3 найефективнішою для побудованої моделі;

- *вперше* проведено моделювання експериментів на невеликій кількості зображень при додаванні до базової моделі блоків детектування групи PFNB, які об'єднують дрібні особливості шарів нейронної згорткової мережі, що

збільшує на невеликому наборі даних усереднене значення mAP та при їх одночасному використанні розмір моделі і кількість параметрів зменшується;

– *вперше* проведено моделювання експериментів на покращеній моделі YOLO v9 P, яка відрізняється від базової моделі YOLO v9 функцією втрат, методом злиття та модифікованою архітектурою блоку розпізнавання, що на невеликому наборі даних дозволило отримати покращення усередненого значення mAP на 7,7% і AP від 2,5% до 14,1%.

Практичне значення одержаних результатів полягає у створенні програмного засобу інформаційної технології розпізнавання та відстеження об'єктів в режимі реального часу.

Результати роботи отримали впровадження в освітній процес кафедри математичного забезпечення ЕОМ факультету прикладної математики Дніпровського національного університету імені Олеся Гончара при викладанні дисциплін «Алгоритми аналізу та методи опрацювання зображень» «Системи штучного інтелекту», «Глибинне навчання» та виконанні кваліфікаційних робіт здобувачів.

Розроблені в дисертаційній роботі технології та програмні засоби отримали впровадження у діяльності ТОВ «КАНЬОН ІНЖИНІРІНГ», зокрема у процесі розробки проєкту «FridgeEye», що дозволяє проводити розпізнавання об'єктів у реальному часі та використовувати в інтелектуальних системах, що є обмеженими за ресурсами.

Наукові результати дослідження є внеском у розвиток архітектурних рішень задачі виявлення, розпізнавання та відстеження об'єктів у режимі реального часу.

В якості можливих напрямків продовження дослідження можна відмітити актуальність розроблення програмного забезпечення для повністю та напів автономних БПЛА.

Ключові слова: інформаційна технологія, виявлення об'єктів, БПЛА, розпізнавання об'єктів, YOLO, зображення, згорткові нейронні мережі, моделі

згорткових нейронних мереж, нейронна мережа, архітектура CNN, глибоке навчання, машинне навчання, Faster R-CNN.

ANNOTATION

Syzonenko O.D. Development of technology and software tools for detecting and recognizing objects in real time – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in Knowledge Area 12 Information Technology, specialty 121 Software Engineering – Oles Honchar Dnipro National University, Dnipro, 2024.

The dissertation is devoted to the development of efficient and fast information technology and software tools for detecting, recognizing and tracking objects in real time.

The study reveals the essence of deep learning technologies and aims to analyze the current state and identify methods to improve the operation of a software solution for detecting, recognizing and tracking objects in real time. The introduction defines the purpose, object, subject, and methods of the study. The scientific novelty of the study, theoretical and practical significance of the scientific results, personal contribution, and information on the implementation and testing of the results are disclosed.

The relevance of the research topic is due to the existing contradiction between the complexity of deep learning models when using existing software solutions, on the one hand, and the growing difficulty of interpreting these models in an application with an increase in the speed of the procedure for recognizing and tracking objects in real time, on the other hand. The field of application of unmanned aerial vehicles (UAVs) is constantly growing and there is a need to adapt machine learning algorithms and deep learning models for use on board UAVs, taking into account the limited computing resources and the specifics of organizing the received video data stream. The paper emphasizes the importance of existing architectural solutions for the implementation of convolutional neural networks when using machine learning methods and algorithms. The identified contradiction is overcome by introducing additional conditions for the task of processing a large set of video

stream images that are constantly updated and can contain information about objects of different sizes and shapes for the task of recognition and tracking. The necessity of using an adapted loss function in the proposed software solution to support decision-making based on the observed data is emphasized. The task of fusing features of convolutional neural networks for spatial and temporal flows is relevant in this area.

The purpose of the study is to improve the accuracy of detection, recognition and tracking of objects in real time and to implement the corresponding technology in the form of a software tool for the task as part of intelligent systems.

The object of research is the processes of data processing in the tasks of detecting, recognizing and tracking objects in real time.

The subject of the study is models, algorithms and technologies for using convolutional neural networks to solve the problem of recognizing and tracking an object in real time.

The first chapter of the thesis focuses on the subject area and analysis of modern methods of object detection, recognition, and tracking. Particular attention is paid to the stages of development of neural convolutional networks and deep learning models for analyzing the built architectural solutions and used algorithms. The problems associated with the implementation of intelligent systems in real time, including the processing of large amounts of data, and optimization of the speed of software tools are also considered.

The second section of the study focuses on the justification of the selected methods for detecting, recognizing, and tracking objects. Attention is paid to the stage of preparing datasets (preprocessing) for their further use in the model. The algorithms of the YOLO and Faster R-CNN families are considered. The details of training these algorithms are described and research results are presented. The main attention is paid to the evolution of the YOLO algorithm, which is key to understanding the structure of this algorithm and the methods and approaches used to organize data.

The third section focuses on the implementation of the technology for recognizing and tracking objects in real time, as well as analyzing the results of the developed technology in comparison with existing software solutions. The existing datasets are reviewed for further testing of the algorithms and the issue of data annotation and the formation of an own dataset is discussed. An experimental study on certain datasets (in particular, our own, Kitti and TAIVD) with a small number of images showed that the proposed technology, in the form of a combination of algorithmic solutions from two types of convolutional neural networks, demonstrates an improved recognition accuracy of about 1.8 % over the analogous YOLO technology. The obtained results of observations from the built own dataset with a small number of images have an advantage in model size by about 6 % relative to the TAIVD dataset in the same volume, which may be appropriate for use due to the incorporation of UAV images. Adding the B2-PFNB detection layer to the base model increases the average mAP accuracy by about 4.3%. Each selected strategy for improving the YOLO analog technology improved the effectiveness of real-time object detection, recognition, and tracking to varying degrees.

The internal software logic of the developed technology is built using a component-oriented approach. To implement the computations, libraries such as Pytorch, Ultralytics, NumPy, OpenCV2, and Matplotlib were used. To deploy this technology, the implementation of several integration points is proposed.

The scientific novelty of the research results is as follows:

- for the first time, an architectural solution for building a neural convolutional network for the task of detecting, recognizing, and tracking objects in real time has been developed, which differs from the existing solution in that it uses a larger number of object recognition units of different sizes, which is optimized for the tasks of a specific subject area;

- for the first time, the possibility of using the PFNB block in the developed technology, which is based on the Faster-Net architectural solution, which uses a

multi-scale feature fusion network to demonstrate improved recognition accuracy compared to the basic technology, has been substantiated;

- for the first time, an own dataset was formed to test the developed technology starting from the stage of object recognition in a video stream, which includes objects of different scales of a certain subject area, which confirms the effectiveness of the developed model;

- for the first time, the architecture of a cross-platform library for the implementation of object detection, recognition and tracking technology is proposed, which is a five-scale structure and contains the BiFormer attention mechanism with low computing power, which improves the accuracy of detecting small objects and improves attention to key information on the object map;

- for the first time, comparative experiments were simulated on YOLO v9, which differ in the use of different types of loss functions while keeping other training conditions unchanged, which showed the use of the WIoU v3 regression loss function to be the most effective for the built model;

- for the first time, experiments were simulated when adding to the basic model the detection units of the PFNB group, which combine small features of the layers of the neural convolutional network, which increases the average value of mAP and, when used simultaneously, the model size and the number of parameters decrease;

- for the first time, experiments were simulated on the improved YOLO v9 P model, which differs from the basic YOLO v9 model in the loss function, fusion method and modified architecture of the recognition unit, which allowed to improve the average mAP value by about 7.7% and AP by about 2.5% to 14.1%.

The practical significance of the results obtained is the creation of a software tool for information technology for recognizing and tracking objects in real time.

The results of the work have been implemented in the educational process of the Department of Mathematical Computer Support of the Faculty of Applied Mathematics of Oles Honchar Dnipro National University in teaching the disciplines

“Algorithms of Analysis and Methods of Image Processing”, “Artificial Intelligence Systems”, “Deep Learning” and in the performance of qualification works of applicants.

The technologies and software tools developed in the dissertation have been implemented in the activities of CANON ENGINEERING LLC, in particular in the development of the FridgeEye project, which allows for real-time object recognition and use in intelligent systems that are limited in resources.

The scientific results of the research contribute to the development of architectural solutions for detecting, recognizing, and tracking objects in real time.

Possible areas for further research include the relevance of developing software for fully and semi-autonomous UAVs.

Keywords: information technology, object detection, UAVs, object recognition, YOLO, images, convolutional neural networks, convolutional neural network models, neural network, CNN architecture, deep learning, machine learning, Faster R-CNN.

Список опублікованих праць за темою дисертації

Статті у наукових фахових виданнях України:

1. Федій О.Д., Божуха Л.М. Про підходи визначення місцезнаходження об'єктів. *Науковий журнал «Математичне моделювання»*. 2021. Вип. 2(45). С. 39-46. DOI: [https://doi.org/10.31319/2519-8106.2\(45\)2021.246874](https://doi.org/10.31319/2519-8106.2(45)2021.246874) URL: <http://matmod.dstu.dp.ua/article/view/246874> (фахове видання категорії Б).
2. Сизоненко О.Д., Божуха Л.М. Підвищення точності геолокації об'єкта на цифровому зображенні при використанні комбінованих технологій аналізу даних. *Науковий журнал «Актуальні проблеми автоматизації та інформаційних технологій»*. 2022. Т.26. С. 103-109. DOI: <http://dx.doi.org/10.15421/432213> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/221> (фахове видання категорії Б).
3. Сизоненко О.Д., Божуха Л.М. Методи локалізації об'єктів на основі зображень із використанням комбінації алгоритмів та багатопоточної зв'язки Faster R-CNN. *Актуальні проблеми автоматизації та інформаційних технологій*. 2023. Т.27. С. 164-177. DOI: <http://dx.doi.org/10.15421/432316> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/241> (фахове видання категорії Б).
4. Сизоненко О.Д., Божуха Л.М. Порівняння YOLO V5 та Faster R-CNN для виявлення об'єктів на зображенні в потоковому режимі. *Системні технології*. 2024. 1(150). С. 51-60. DOI: <https://doi.org/10.34185/1562-9945-1-150-2024-05> URL: <https://journals.nmetau.edu.ua/index.php/st/article/view/1523> (фахове видання категорії Б).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Сизоненко О. Д., Божуха Л.М. Виявлення місцезнаходження бпла за допомогою зіставлення зображень з використанням ключових точок. *XXI Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем»*: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2023 рік. Дніпро, 2023, С. 266-267, URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2023/11/mpzis-2023.pdf>.

6. Сизоненко О.Д., Божуха Л.М. Виявлення місцезнаходження об'єктів за допомогою GIS. *XX Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем»*: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2022 рік. Дніпро, 2022, С. 178, URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf>.

7. Федій О.Д., Божуха Л.М. Про алгоритми позиціювання об'єктів в локальній мережі. *XIX Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем»*: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2021 рік. Дніпро, 2021, С. 201, URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2021/11/mpzis_2021.pdf.

8. Сизоненко О.Д., Божуха Л.М. Методи прив'язки зображення до геолокації. *Всеукраїнська науково-методична конференція «Проблеми математичного моделювання»*: тези доповідей Всеукраїнської науково-методичної конференції за 2022 рік. Кам'янське, 2022, С. 84, URL: https://www.dstu.dp.ua/uni/downloads/zbirka_konf_pm.pdf.

9. Сизоненко О.Д., Божуха Л.М. Експериментальні результати встановлення геолокації об'єкта при використанні мережі виявлення об'єктів Faster R-CNN. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)*: тези доповідей XX міжнародної науково-практичної

конференції, Дніпро, 2022, виступ є, без публікації, URL:
https://www.dnu.dp.ua/docs/ndc/2023/Ost_var_programa.pdf.

ЗМІСТ

ВСТУП.....	20
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ВИЯВЛЕННЯ, РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ ОБ'ЄКТІВ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ	28
1.1 Аналіз потреб для розвитку БпЛА.....	28
1.1.1 Аналіз вимог застосування та критерії використання БпЛА.....	30
1.1.2 Проблематика збору даних.....	32
1.2 Аналіз існуючих рішень задачі виявлення та розпізнавання об'єктів..	34
1.2.1 Постановка задачі виявлення та розпізнавання об'єктів.....	37
1.2.2 Методи віднімання фону.....	38
1.2.3 Метод Віоли-Джонса для виявлення об'єктів в режимі реального часу.....	40
1.2.4 AdaBoost для підвищення ефективності алгоритмів класифікації	41
1.2.5 Алгоритм Fast R-CNN.....	43
1.2.6 Модель Faster R-CNN.....	48
1.2.7 Опис методу SSD.....	50
1.2.8 Алгоритм YOLO.....	52
1.2.9 Бібліотека OpenCV.....	58
1.3 Аналіз існуючих рішень задачі відстеження об'єктів.....	59
1.3.1 Задача відстеження об'єктів.....	61
1.3.2 Безперервне адаптивне відстеження середнього зсуву.....	62
1.3.3 Опис TransMOT.....	63
1.3.4 Метод відстеження BYTE Track.....	65
1.4 Висновки до розділу 1.....	66
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ОБЧИСЛЮВАЛЬНИХ АСПЕКТІВ МЕТОДІВ	

ВИЯВЛЕННЯ, РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ.....	68
2.1 Виявлення об'єктів у наборі даних Kitti при застосуванні YOLO та Faster R-CNN.....	70
2.1.1 Опис початкових даних.....	71
2.1.2 Опис набору даних Kitti.....	71
2.1.3 Порівняльна оцінка алгоритмів сімейства YOLO та Faster R-CNN.....	73
2.1.4 Ключові показники для оцінки.....	75
2.1.5 Деталі навчання для Faster R-CNN.....	77
2.1.6 Деталізація етапів навчання для YOLOv3 та YOLOv5.....	79
2.1.7 Результати досліджень та порівняльний аналіз.....	80
2.1.8 Аналіз часу виконання алгоритмів моделей.....	85
2.2 Моделі та реалізації алгоритму YOLO.....	86
2.2.1 Алгоритмічна основа.....	87
2.2.2 Еволюція архітектури мережі YOLO.....	90
2.2.3 Комплексна функція втрат.....	102
2.3 Опис власної технології YOLO v9 P.....	104
2.3.1 Покращення функції втрат.....	106
2.3.2 Ефективний механізм уваги.....	110
2.3.3 Багатомасштабна мережа об'єднання ознак.....	114
2.4 Алгоритми препроцесингу зображень.....	120
2.4.1 Видалення шуму.....	120
2.4.2 Видалення ефекту розмиття.....	122
2.4.3 Збільшення контрастності.....	126
2.5 Алгоритми BYTE Track для відстеження багатьох об'єктів.....	129
2.5.1 Алгоритми методу BYTE Track.....	130

2.5.2 Фільтр Калмана.....	131
2.6 Висновки до розділу 2.....	137
РОЗДІЛ 3. РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСОБУ РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ.....	139
3.1 Огляд існуючих наборів даних (dataset).....	139
3.2 Формування власного датасету.....	143
3.2.1 Вибір програмного інструментарію для формування набору даних.....	144
3.2.2 Порівняльний аналіз Label Studio та Roboflow для анотування даних.....	146
3.3 Обґрунтування технологій та середовища для розроблення програмного забезпечення.....	148
3.4 Результати експериментів.....	150
3.4.1. Порівняльні експерименти з функціями втрат різного вигляду.	150
3.4.2. Порівняння результатів модифікованого алгоритму YOLO v9 Р з базовим YOLO v9.....	151
3.4.3. Додавання блоку ViFormer.....	154
3.4.4. Експерименти з порівнянням впровадження різних стратегій покращення.....	155
3.5 Висновки до розділу 3.....	158
ВИСНОВКИ.....	160
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	162
ДОДАТОК А Список публікацій здобувача.....	169
ДОДАТОК Б Акт впровадження.....	172

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БпЛА	– безпілотний літальний апарат
IoU	– коефіцієнт перетину над об'єднанням
AP	– середня похибка (англ. average precision)
GPS	– супутникова навігація
Faster R-CNN	– модель відноситься до групи розпізнавання за областю, що розпізнає об'єкти у два етапи
mAP	– усереднена похибка (англ. mean average precision)
TAIVD	– Traffic Aerial Images for Vehicle Detection (dataset)
PFNB	– Perception FasterNet Block
YOLO	– метод розпізнавання об'єктів (англ. You Only Live Once, YOLO)
WIoU	– функція втрат (англ. Wise IoU)
ГІС	– геоінформаційна система

ВСТУП

Актуальність теми. З кожним днем дедалі більше зростає кількість сфер діяльності, у яких суттєву допомогу надають БпЛА [1]. А саме: зрошення сільськогосподарських угідь, гасіння пожеж, доставка вантажів, пошук та порятунок людей, моніторинг районів, і навіть під час бойових дій. Завдяки своїм можливостям, БпЛА стають ефективними інструментами для збору інформації та аналізу в реальному часі. Однак, точне та швидке виявлення, розпізнавання та відстеження об'єктів у відеопотоці даних в режимі реального часу залишається викликом, який потребує подальшого вдосконалення. Існує багато підходів, методів, алгоритмів та технологій виявлення об'єктів зображення або відеопотоку та знаходження місцерозташування об'єктів з борту БпЛА [2], наприклад, за допомогою GPS або при використанні комбінованих технологій [3-6]. Але в сучасних реаліях слід пам'ятати, що на точність і надійність GPS можуть впливати фактори навколишнього середовища, такі як перешкоди від будівель, дерев та інших об'єктів, також сигнали GPS можуть блокувати під час спеціальних операцій. Крім того, на точність GPS може впливати потужність сигналу, на яку впливає висота дрона та відстань до передавача.

БпЛА мають низку переваг перед наземними засобами: велика область покриття територій, можливість приземлення у будь-яку точку місцевості, висока швидкість польоту та, як наслідок, оперативність доставки вантажів незалежно від дорожніх обставин та інше.

Для задоволення зростаючих вимог до цього класу авіаційної техніки вченими та інженерами виконуються дослідження, спрямовані як на покращення льотних характеристик апаратів, так і на вдосконалення їх технічного, інформаційного та програмного забезпечення.

Ефективне виявлення, розпізнавання та відстеження об'єктів у відеопотоці є ключовою задачею, що дозволяє БпЛА виконувати різноманітні

завдання в автономному та напівавтономному режимах. Отже, розроблення та дослідження ефективності нових універсальних алгоритмів, технологій та програмних засобів, що органічно поєднують властивості класичних методів, алгоритмів машинного навчання та моделей глибинного навчання і на їх основі побудованих відповідних програмних засобів, – є актуальною, як в теоретичному так і в прикладному застосуванні.

Зв'язок роботи з науковими програмами, планами, темами.

Запропоновані технологія та модель створені в рамках досліджень наукової школи «Інформаційні технології обробки статистичних даних» на кафедрі математичного забезпечення ЕОМ Дніпровського національного університету імені Олеся Гончара.

Дисертаційна робота виконана відповідно з поточними та перспективними планами наукової та науково-технічної діяльності Дніпровського національного університету імені Олеся Гончара для подальшого розвитку інженерії програмного забезпечення.

Метою дисертаційної роботи є підвищення точності виявлення, розпізнавання та відстеження об'єктів в режимі реального часу та реалізації відповідної технології у вигляді програмного засобу.

Результати цього дослідження можуть бути використані для подальшого вдосконалення систем автоматичного розпізнавання та відстеження об'єктів в режимі реального часу при покращенні функціональності програмного забезпечення інтелектуальних систем, зокрема БпЛА.

Досягнення вказаної мети дослідження забезпечується виконанням наступних завдань:

- провести аналіз існуючих технічних та програмних рішень виявлення об'єктів у відеопотоці даних та подальшого їх розпізнавання та відстеження;
- провести аналіз алгоритмів та структур даних для врахування обмеженості на обчислювальні ресурси та специфіку організації отриманого відеопотоку даних;

- виконати препроцесінг зображень для навчання моделі та вирішення задачі розпізнавання та відстеження об'єктів відеопотоку даних з БпЛА;
- удосконалити існуюче архітектурне рішення реалізації згорткової нейронної мережі при використанні методів та алгоритмів машинного навчання;
- удосконалити умову існуючого алгоритму щодо об'єктів різного розміру та форми задачі опрацювання набору зображень відеопотоку даних;
- розробити програмне рішення щодо підключення адаптованої функції втрат для підтримки прийняття рішень на основі спостережених даних;
- розробити програмне рішення щодо підключення методу злиття ознак згорткових нейронних мереж для просторового та часового потоків;
- реалізувати технологію розпізнавання та відстеження об'єктів з використанням сучасних технологій комп'ютерного зору та штучного інтелекту;
- провести експеримент та оцінити ефективність запропонованої технології розпізнавання та відстеження об'єктів відеопотоку даних з БпЛА.

Об'єктом дослідження є процеси оброблення даних в задачах виявлення, розпізнавання та відстеження об'єктів у режимі реального часу.

Предметом дослідження є моделі, алгоритми та програмні засоби використання згорткових нейронних мереж для вирішення задачі розпізнавання та відстеження об'єкта.

Методи дослідження: методи системного та порівняльного аналізу; методи та технології інженерії програмного забезпечення; структурний аналіз та синтез аналізу умов, що впливають на функціонування окремих модулів моделі та побудови архітектури програмного рішення; методи препроцесінгу; методи машинного навчання; методи побудови нейронних згорткових мереж; методи імітаційного моделювання при апробації запропонованої технології.

Наукова новизна одержаних результатів:

1. Вперше розроблено архітектурне рішення побудови нейронної згорткової мережі задачі виявлення, розпізнавання та відстеження об'єктів в режимі реального часу, що відрізняється від існуючого рішення тим, що використовує більшу кількість блоків розпізнавання об'єктів різного розміру, яке є оптимізоване для задач конкретної предметної області.

2. Вперше обґрунтовано можливість використання в розробленій технології PFNB-блоку, який базується на архітектурному рішенні Faster-Net, що використовує багатомасштабну мережу об'єднання ознак для демонстрації покращеної точності розпізнавання у порівнянні з базовою технологією.

3. Вперше сформований власний набір даних для апробації розробленої технології починаючи з етапу розпізнавання об'єктів у відеопотоці, який включає об'єкти різного масштабу визначеної предметної області, що підтверджує ефективність розробленої моделі.

4. Вперше запропоновано архітектуру кроссплатформної бібліотеки для реалізації технології виявлення, розпізнавання та відстеження об'єктів, яка є п'ятимасштабною структурою і містить механізм уваги ViFormer з малою обчислювальною потужністю (зменшує розмір моделі на 0,4 МБ на невеликому наборі даних), що дозволяє покращити точність виявлення малих об'єктів та покращує увагу до ключової інформації на карті об'єктів і збільшує mAP50 на 0,5%.

5. Вперше проведено моделювання порівняльних експериментів на YOLO v9 на невеликому наборі даних, які відрізняються використанням різних видів функцій втрат при зберіганні інших умов навчання незмінними, що показало використання функції регресійних втрат WIoU v3 найефективнішою для побудованої моделі і значення mAP моделі при використанні WIoU v3 на 0,7% вище, ніж при використанні CIoU.

6. Вперше проведено моделювання експериментів при додаванні до базової моделі блоків детектування групи PFNB, які об'єднують дрібні особливості шарів нейронної згорткової мережі, що збільшує на невеликому

наборі даних значення mAP на 4,3% та при їх одночасному використанні розмір моделі і кількість параметрів зменшується зменшується на 5% (1 MB) і кількість параметрів зменшується більше ніж на 7,2% ($0,8 * 10^6$).

7. Вперше проведено моделювання експериментів на покращеній моделі YOLO v9 P, яка відрізняється від базової моделі YOLO v9 функцією втрат, методом злиття та модифікованою архітектурою блоку розпізнавання, що на невеликому наборі даних дозволило отримати покращення значень mAP на 7,7% і AP від 2,5% до 14,1%..

Практичне значення одержаних результатів полягає у створенні програмного засобу інформаційної технології розпізнавання та відстеження об'єктів в режимі реального часу.

Розроблене програмне забезпечення для вирішення задачі виявлення, розпізнавання та відстеження об'єктів у режимі реального часу дозволяє адаптувати розроблені рішення до існуючих систем ідентифікації об'єктів.

Результати роботи отримали впровадження в освітній процес кафедри математичного забезпечення ЕОМ факультету прикладної математики Дніпровського національного університету імені Олеся Гончара при викладанні дисциплін «Алгоритми аналізу та методи опрацювання зображень» «Системи штучного інтелекту», «Глибинне навчання» та виконанні кваліфікаційних робіт здобувачів.

Розроблені в дисертаційній роботі технології та програмні засоби отримали впровадження у діяльності ТОВ «КАНЬОН ІНЖИНІРІНГ», зокрема у процесі розробки проекту «FridgeEye», що дозволяє проводити розпізнавання об'єктів у реальному часі та використовувати в інтелектуальних системах, що є обмеженими за ресурсами.

Наукові результати дослідження є внеском у розвиток архітектурних рішень задачі виявлення, розпізнавання та відстеження об'єктів у режимі реального часу. Дослідження може бути використане як основа для подальших наукових досліджень у галузі комп'ютерного зору та штучного інтелекту.

В якості можливих напрямків продовження дослідження можна відмітити актуальність розроблення програмного забезпечення для повністю та напів автономних БПЛА.

Особистий внесок здобувача. Дисертаційна робота є самостійною науковою працею, в якій висвітлені власні ідеї і розробки автора, що дозволили вирішити поставлені завдання. Основні положення, висновки, результати дослідження і рекомендації, що містяться у дисертаційній роботі та виносяться на захист, отримано здобувачем особисто. Використані в дисертації ідеї, положення чи гіпотези інших авторів мають відповідні посилання і використані лише для підкріплення ідей здобувача.

За темою дисертації опубліковано 9 наукових праць. З праць виконаних зі співавторами, на захист виносяться лише результати, отримані особисто здобувачем. У наукових працях, опублікованих у співавторстві, автору належать:

- у роботі [2] виконаний аналіз сучасних технологій та програмних засобів щодо визначення місцезнаходження об'єкта; запропоновано підходи щодо формування стратегії обрання принципів архітектури мережі при проєктуванні системи для подальшого розроблення технології усунення всіх можливих факторів, які впливають на визначення місця розташування з максимальною точністю; проведено дослідження і надані результати порівняння використання різних підходів;

- у роботі [3] приділено увагу оцінці GPS-розташування зображення з фоновим зображенням вулиць шляхом пошуку відповідних зображень у довідковій базі даних зображень та використання алгоритмів порівняння; розроблено систему за допомогою Faster R-CNN для виявлення будівель за запитом; для визначення об'єктів обрано алгоритм k-найближчих сусідів з використанням сіамської згорткової нейронної мережі, враховано позитивні та негативні пари зображень за збігом; оцінено запропоновану структуру на наборі зображень з різними характеристиками зйомки;

– у роботі [11] приділена увага процесу розроблення програмного продукту з використанням багатопоточної зв'язки Faster R-CNN та методу різниці кадрів з використанням технології розпаралелювання;

– у роботі [32] описано розроблене програмне забезпечення для порівняння двох моделей YOLO v5 та Faster R-CNN щодо вирішення задачі виявлення об'єктів; наведено результати навчання і валідації на експериментальному наборі даних;

– у тезах доповідей [8] запропоновано систему, яка замінює сигнал GPS, поєднуючи метод яскравості пікселів, метод різниці кадрів та Faster R-CNN;

– у тезах доповідей [4] запропоновано метод покращення виявлення об'єктів з використанням набору абсолютних значень, отриманих із баз даних ГІС;

– у тезах доповідей [5] для алгоритма позиціонування об'єктів в локальній мережі запропоновано модель розташування об'єктів при використанні підходів визначення локації в режимі реального часу;

– у тезах доповідей [6] запропоновано методи, що прив'язують зображення до певного розташування: за метаданими, реверсивним пошуком, за об'єктами-якорями на фоні, за описом;

– у тезах доповідей [12] приділено увагу програмному продукту з використанням методу Faster R-CNN для виявлення об'єктів та проведено експеримент щодо встановлення геолокації об'єкта.

Апробація матеріалів дисертації. Основні результати дисертаційного дослідження обговорено на міжнародних та всеукраїнських наукових конференціях, зокрема на: «Математичне та програмне забезпечення інтелектуальних систем», (MPZIS-2023) (м. Дніпро, 2023 р.) [8]; «Математичне та програмне забезпечення інтелектуальних систем», (MPZIS-2022) (м. Дніпро, 2022 р.) [4]; «Математичне та програмне забезпечення інтелектуальних систем», (MPZIS-2021) (м. Дніпро, 2021 р.) [5]; Всеукраїнська науково-методична конференція «Проблеми математичного моделювання» (м.

Кам'янське, 2022 р.) [6]; «Наукова конференція за підсумками науково-дослідної роботи ДНУ ім.О. Гончара за 2022 роки. Математичні та програмні засоби обробки і аналізу даних» (м. Дніпро, 2022 р.) [12]. Результати дисертації також обговорювалися на наукових семінарах факультету прикладної математики Дніпровського національного університету імені Олеся Гончара.

Публікації. Основні наукові результати дисертації викладено у 9 публікаціях, серед яких 4 статті в періодичних наукових журналах і збірниках наукових праць категорії Б, 5 публікацій у працях і матеріалах наукових конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Повний обсяг дисертації складає 172 сторінки, в тому числі 142 сторінок основного тексту, 14 таблиць і 58 рисунків, список використаних джерел із 47 найменувань і 2 додатки.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ВИЯВЛЕННЯ, РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ ОБ'ЄКТІВ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

В даному розділі буде розглянуто актуальність дослідження дисертаційної роботи та обґрунтовано практичну необхідність розроблення технології для вирішення описаної проблеми при застосуванні специфіки задач комп'ютерного зору з використанням БпЛА.

На основі наведеного опису прикладних задач застосування моделей комп'ютерного зору в складі технологій та програмного забезпечення БпЛА, необхідно виокремити задачі та методи алгоритмів машинного та/або глибокого навчання для забезпечення необхідної ефективності та точності розпізнавання.

Для вирішення проблеми вибору побудованих архітектурних рішень, технологій та алгоритмів, буде приділена увага етапам розвитку нейронних мереж та моделей глибокого навчання при врахуванні змін, слабких та сильних сторін для кожного з оглянутих програмних рішень.

Необхідно сформулювати вимоги для технології виявлення та розпізнавання об'єктів у режимі реального часу, які пов'язані з реалізацією інтелектуальних систем у режимі реального часу при опрацюванні великого обсягу даних та оптимізацію швидкості роботи програмних засобів.

1.1 Аналіз потреб для розвитку БпЛА

Виконання цього завдання вимагає надійності системи, високої точності алгоритмів та швидкості роботи програмних засобів, оскільки БпЛА використовуються в різних сферах, зокрема військовій.

В останні роки БпЛА змогли покрити більшість завдань, які раніше виконувалися пілотованими літальними апаратами. Оскільки технологія БпЛА

продовжує розвиватися, кількість безпілотників в країнах світу зростає з кожним роком, і згідно з даними, що показує глобальні комерційні дрони за рік і статистику продажів [7], до 2025 року в світі буде приблизно 2679000 БпЛА, при цьому обсяг ринку становить приблизно 12,6 мільярдів доларів США (рисунок 1.1). Отже, це свідчить про швидкий розвиток БпЛА та його поширення.

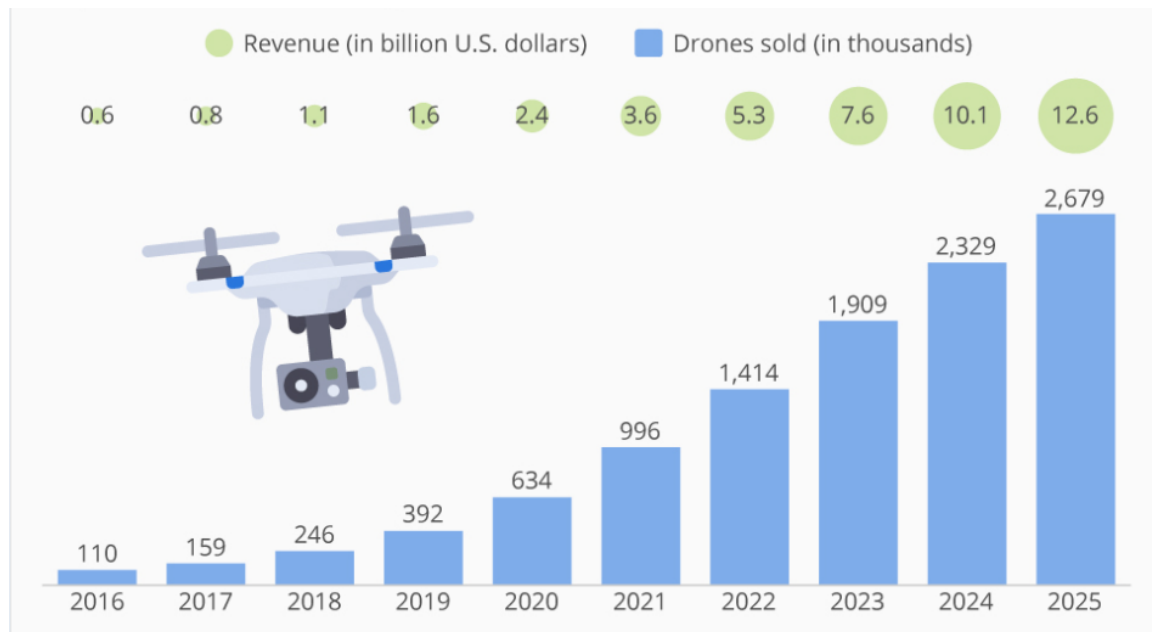


Рисунок 1.1 – Статистика кількості реалізованих комерційних БпЛА з передбаченням до 2025 року [7]

Але швидкий ріст в сфері БпЛА збільшує кількість проблемних задач, які пов'язані з розпізнаванням рухомих об'єктів. Факторами, які можуть впливати на появу додаткових умов можуть бути зміна ракурсу, мінливість освітлення, різні варіації у розмірі та вигляді об'єктів, перекриття об'єктів, шум, тощо. Згідно з дослідженням, проведеним у 2020 році, понад 70% помилок у розпізнаванні об'єктів на зображеннях БпЛА пов'язані з недостатньою точністю алгоритмів.

Актуальною задачею є побудова алгоритмів, які можуть працювати в режимі реального часу. Завдяки збільшенню потужності обчислювальних

пристроїв та застосуванню оптимізованих алгоритмів, можливо досягти покращених результатів при роботі з відеопотоком даних БпЛА в режимі реального часу. Крім того, існують проблеми, що пов'язані з універсальністю та масштабованістю алгоритмів розпізнавання об'єктів. Виникає потреба в розробленні методів, які здатні розпізнавати різні типи об'єктів за розміром та формою, включаючи людей, транспортні засоби, будівлі тощо. Ефективність роботи алгоритму в умовах зміни масштабу та великого обсягу даних теж потребує уваги.

У зв'язку зі зростанням кількості БпЛА стають дедалі важливішими проблеми безпеки та приватності. Виникає потреба в алгоритмах, які можуть ефективно виявляти загрози та захищати приватні дані та при цьому забезпечувати безпеку операторів та інших учасників.

Виявлення, розпізнавання та відстеження об'єктів в режимі реального часу для БпЛА є актуальною та складною задачею в галузі комп'ютерного зору. Для досягнення високої точності розпізнавання об'єктів в режимі реального часу, широко використовуються згорткові нейронні мережі. Ці методи дозволяють автоматично вивчати характеристики об'єктів з великого обсягу даних і формувати точні передбачення на нових зображеннях або відео.

Враховуючи вищезазначені проблеми та виклики, розроблення ефективних методів виявлення та розпізнавання об'єктів для БпЛА є актуальною та важливою задачею, яка вимагає подальших досліджень та розвитку.

1.1.1 Аналіз вимог застосування та критерії використання БпЛА

Актуальним є питання застосування БпЛА в різних галузях. Розпізнавання та відстеження об'єктів є необхідною складовою для виконання різних завдань в галузі безпеки, таких як виявлення загроз, спостереження та збір розвідувальної інформації. У сфері транспорту та логістики безпілотні

літальні апарати можуть бути використані для ефективного моніторингу та керування логістичними процесами. Розпізнавання та відстеження рухомих об'єктів є ключовим фактором для автоматичного виявлення та ідентифікації об'єктів на зображеннях, що дозволяє покращити ефективність та безпеку логістичних операцій.

БпЛА здобувають все більше значення у сільському господарстві завдяки їхньому потенціалу для автоматизації: моніторинг і обстеження сільськогосподарських культур (БпЛА забезпечують можливість візуального контролю за станом полів та рослин, що дозволяє фермерам вчасно виявляти проблеми, такі як хвороби, шкідники та стресові фактори, і приймати відповідні заходи); полив та дозування (БпЛА з системами для поливу та розпилення добрив, які дозволяють рівномірно та ефективно застосовувати ресурси та підвищувати врожайність); попередження та діагностика кризових ситуацій (БпЛА можуть швидко оцінювати наслідки природних лих, таких як засуха, повінь або град, і допомагати в реагуванні на них, а також попереджати фермерів про можливі загрози).

Актуальність розпізнавання об'єктів для БпЛА є очевидною, оскільки це є ключовою технологією для покращення безпеки, ефективності та функціональності безпілотної авіації. Зростання ринку БпЛА як у світі, так і в Україні, підкреслює необхідність розвитку нових методів та алгоритмів оперативного моніторингу об'єктів в режимі реального часу.

Використання виявлення та розпізнавання об'єктів у сфері БпЛА має свою специфіку тому, що отримані фото- та відео- дані з БпЛА залежать від багатьох факторів, починаючи від технічних характеристик самого апарату до середовища його польоту, під час якого були зібрані дані.

Різні типи БпЛА мають різні технічні характеристики (рисунок 1.2). Наприклад, одногвинтові БпЛА можуть нести важке корисне навантаження, але вони мають складну механічну конструкцію, що призводить до високої вартості. Багатогвинтові БпЛА є поширеним типом, оскільки їх можна

використовувати професійно та вони мають можливість зависати або рухатися вздовж заданої цілі. Іншим критерієм класифікації БпЛА, окрім аеродинамічної концепції, може бути їх стандарт автономії. Коли пілот дає сигнал на кожен привід літального апарату, він може бути віднесений до радіо-операційного (наприклад, FPV апарати). Радіокеровані апарати (наприклад, апарати серії DJI Mavic) демонструють другий ступінь автономності, в яких безпека польоту залежить від бортового автоматичного контролера та наземний оператор може надавати бортовому контролеру команди швидкості та орієнтації. Без втручання людини автономні апарати можуть виконувати певний план польоту.

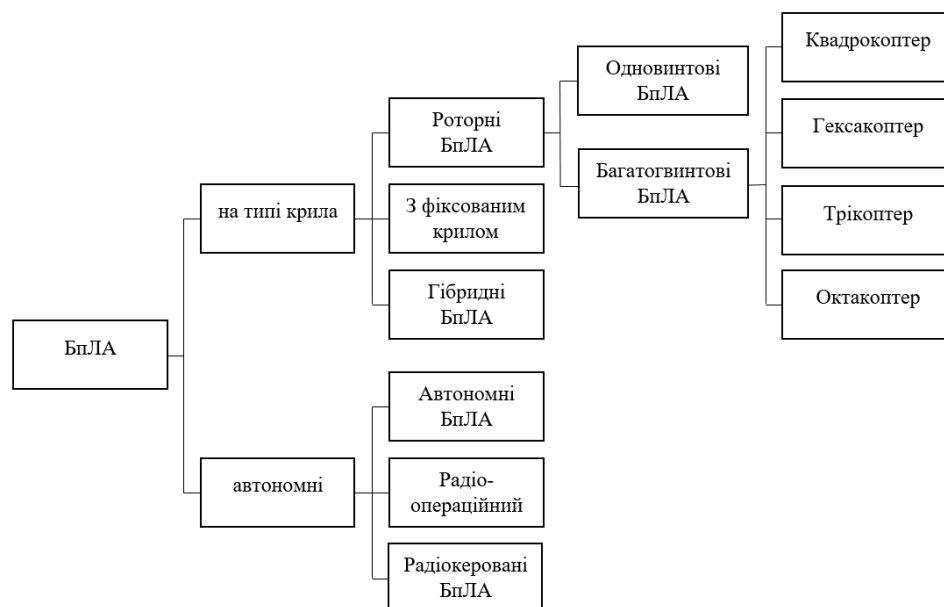


Рисунок 1.2 – Класифікація БпЛА

1.1.2 Проблематика збору даних

Використання методів та алгоритмів комп'ютерного зору у сфері БпЛА, зокрема алгоритми розпізнавання та відстеження багатьох об'єктів, має окрему проблематику спричинену способом збору даних. Проблеми вирішуються як технічними засобами, так і алгоритмічно, в залежності від обґрунтування мети розроблення технології, основні з яких наведено нижче.

Динамічна зміна сцени є наслідком руху БпЛА на значній швидкості та його маневрування. На зображенні даний тип проблеми може відображатися розмиттям внаслідок руху, що спотворює якість. Дана проблема може вирішуватися технічно – шляхом заміни камери на нову з збільшеною частотою зйомки або алгоритмічно – використовуючи методи видалення розмиття. Подібною за природою є проблема рухомих об'єктів відносно сцени, що приводить до розмиття окремих об'єктів, у вирішенні якої найбільший успіх мають мережі глибокого навчання.

Проблема зміни ракурсу камери (представленим кутами до вертикалі та до горизонту) приводить до зменшення точності розпізнавання об'єктів та погіршення деяких методів відстеження. Наприклад, зміна ракурсу призводить до варіативності вигляду одного об'єкту (проекція згори відрізняється від бокової проекції). Для вирішення цієї проблеми використовують більшу вибірку для навчання, яка може бути отримана з різних ракурсів.

Проблему розпізнавання малих об'єктів спричиняє велика відстань між БпЛА та об'єктами сцени при завданні якнайбільшого покриття сцени (тобто без використання збільшення зображення). Складність даної проблеми базується на малій інформативності об'єктів відносно загальної сцени, що робить задачу розпізнавання важчою ніж при фронтально-паралельному зніманні (наприклад, камери над дорогами).

Погані погодні умови також спричиняють ряд проблем, таких як недостатня контрастність кадру, різні типи розмиття, тощо. Проблема освітленості, що приводить до нехватки контрастності зображення, може бути спричинена порою доби або року та погодними умовами, наприклад в сонячний день зображення стає «засліпленим», а ввечері «темним». Цю проблему зазвичай вирішують технічно – збільшенням чутливості камери, алгоритмічно – методами підвищення контрастності. Туман, хмарність

можуть також призвести до проблеми розмиття, а інколи це може статися через неправильне налаштування фокусу камери.

На додаток до цих труднощів, дослідження щодо виявлення об'єктів на відеоданих з БпЛА також стикаються з проблемою упередженості набору даних. Щоб уникнути цієї проблеми, набір даних повинен мати анотацію, яка відображає реальні застосування. Тому не дивно, що моделі для розпізнавання об'єктів, навчені на традиційних зображеннях, не підходять для відеоданих з БпЛА.

На додаток ще можна отримати проблему фактичного збору даних в режимі реального часу для проведення експериментів щодо апробації алгоритмів виявлення та розпізнавання об'єктів. В умовах воєнного стану саме з цією проблемою виникали складнощі в організації збору даних для дисертаційного дослідження, що спонукало до використання інших підходів організації набору анотованих даних.

1.2 Аналіз існуючих рішень задачі виявлення та розпізнавання об'єктів

Розпізнавання об'єктів є завданням комп'ютерного зору, метою якого є пошук одного та/або групи об'єктів на зображенні або у відеоряді за виділеними істотними ознаками.

Алгоритми завдання комп'ютерного зору спрямовані на розуміння змісту зображення, передбачення просторового розташування, виявлення рамки об'єктів на зображенні (переважно прямокутником) та класифікування об'єктів за визначеними категоріями.

За останні роки, як наслідок розвитку інформаційних технологій та алгоритмів виявлення та розпізнавання об'єктів, з'явилося багато нових моделей і їх модифікацій. Сучасні побудовані алгоритми розпізнавання об'єктів відрізняються між собою тим, що використані різні методи або їх

комбінації, функціональні вимоги задачі (наприклад, від розпізнавання в реальному часі і до мультимасштабного розпізнавання), можливості обчислювальної техніки, на якій передбачено тренування та апробація алгоритмів, тощо.

Підходи щодо побудови моделей за алгоритмічною базою можна поділити на дві групи: на основі класичних методів оброблення зображень (методами машинного навчання) та на основі нейронних мереж глибокого навчання (рисунк 1.3).

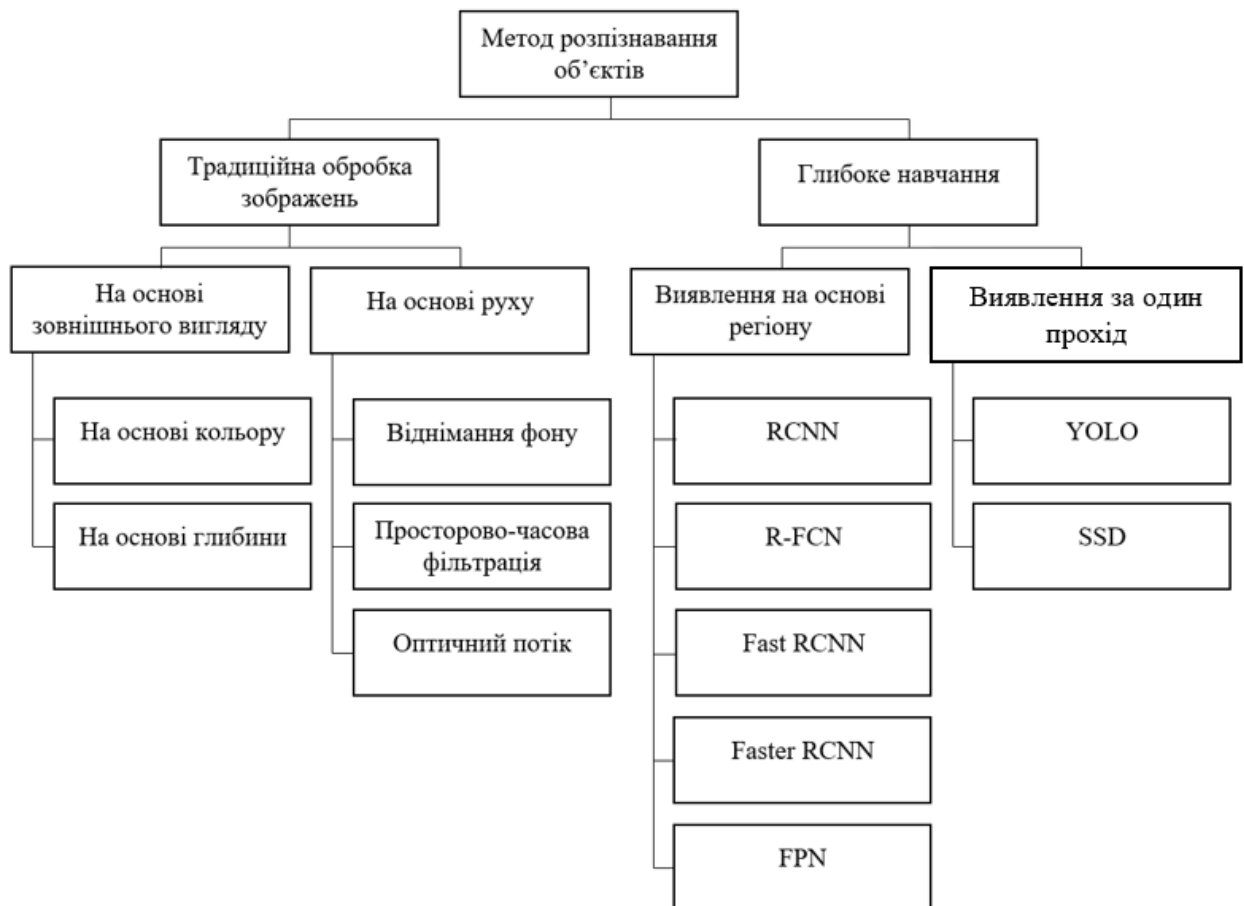


Рисунок 1.3 – Класифікація методів розпізнавання об'єктів

Для розпізнавання об'єктів на зображенні застосовують дві стратегії: моделювання фону і моделювання об'єкта. Вибір стратегії залежить від умов отримання зображення. Моделювання фону застосовне тільки для “ідеальних” умов зйомки. При виборі стратегії моделювання об'єкта, яка має більш загальний підхід, можна застосовувати різні методи для пошуку об'єкта на

зображенні. Ефективність використання кожної з цих стратегій залежить від багатьох умов, зокрема вибору методів та алгоритмів опрацювання зображень.

Найпростішими методами виділення об'єкта на зображенні є колірні фільтри. Такі методи застосовуються, якщо об'єкт суттєво виділяється на фоні. Виділення країв та контурний аналіз можуть бути корисними у випадках, якщо об'єкт досить складний і добре виділяється на фоні. Використання таких методів надає змогу переходити від роботи з зображенням до роботи з об'єктами на цьому зображенні.

Далі вже може з'явитися можливість перевірки наявності на зображенні певних геометричних форм. Метод співставлення зі шаблоном (template matching) полягає у пошуку на зображенні ділянок, які співпадають з зображенням шуканого об'єкта. При умові, що зображення об'єкта повернуте чи масштабоване відносно шаблону, цей метод не є ефективним. Для таких випадків запропоновані методи, які базуються на використанні особливих (опорних) точок (reference points). Опорні точки є особливими характеристиками шуканого об'єкта на зображенні. Ці характеристики дозволяють співставити об'єкт сам з собою або зі схожими класами об'єктів. Існує декілька способів виділення особливих точок: на сусідніх кадрах, через великі проміжки часу та при різному освітленні, при повороті зображення та ін.

Найскладнішими випадками задачі розпізнавання є пошук об'єктів визначеного класу. В таких випадках задачу виявлення і розпізнавання можна вирішити за допомогою побудови класифікатора на основі машинного навчання, який складається з метода виділення особливостей (feature extractor) та власне класифікатора. Вибір методу виділення особливостей залежать від поставленої задачі. Для одного класу задач це може бути навчання на позитивних і негативних наборах даних з зображеннями, для інших – це виділення кластерів дескрипторів особливих точок і створення словника дескрипторів. Не зважаючи на велику кількість методів розпізнавання

об'єктів, не існує універсального набору для всіх умов виконання задачі розпізнавання об'єктів. Задача розпізнавання може бути вирішена лише за конкретних умов та визначенні набору методів та алгоритмів.

1.2.1 Постановка задачі виявлення та розпізнавання об'єктів

Розпізнавання об'єктів є підзадачею групи задач виявлення об'єктів в сфері комп'ютерного зору. Формальну постановку задачі розпізнавання об'єктів можна виразити як комбінацію задачі локалізації (просторового розташування) та задачі класифікації об'єктів на зображенні відеоряду (рисунок 1.4).



Рисунок 1.4 – Результат для задач класифікації зображення, локалізації та розпізнавання об'єкту

Вхідними даними для локалізації об'єктів є зображення I , а вихідними є набір обмежувальних прямокутних рамок $B_1, B_2 \dots B_n$, який представлений координатами об'єкту та розміром:

$$B: \{x, y, w, h\}, \quad (1.1)$$

де

x, y – координати верхнього лівого кута прямокутника обмежувальної рамки,

w, h – ширина та висота обмежувальної рамки.

Задача класифікації зображення. Об'єкт O є членом одного з визначених класів $C_i, i = \{1, 2, \dots, n\}, n \in N$. Вхідними даними є зображення I , на якому представлено лише один об'єкт O , або його відсутність. Вихідними даними є клас C_i з попередньо визначеної множини класів $C_1, C_2 \dots C_n$.

Задача розпізнавання об'єкту полягає в локалізації об'єктів на зображенні з визначенням їх приналежності до певного класу. Вхідними даними слугує зображення I , на якому знаходиться множина об'єктів $\{O_1, O_2 \dots O_m\}, m \in N$. На виході очікується множина наборів з 2 елементів для кожного об'єкту зображення:

$$O_i : (B_i, C_j), i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}, m, n \in N \quad (1.2)$$

де

m – кількість об'єктів на зображенні,

n – кількість класів.

1.2.2 Методи віднімання фону

Для детектування руху широко використовуються методи віднімання фону, що застосовується в більшості традиційних задач оброблення зображень.

Метод віднімання фону фокусується на розпізнаванні в реальному часі об'єктів переднього плану за відеорядом, шляхом віднімання пікселів фонових сцен (рисунок 1.5). Цей процес складається з трьох етапів: ініціалізація фону, підтримка фону та класифікація пікселів фону [8].

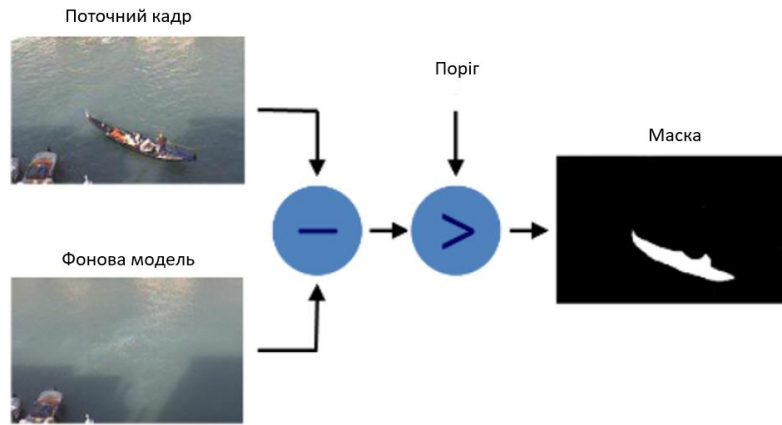


Рисунок 1.5 – Приклад роботи методу віднімання фону

Початковим кроком є обчислення часової різниці кадрів. Різниця кадрів у певний момент часу t обчислюється за формулою:

$$FD_t = |p_t(x, y) - p_{t-1}(x, y)|, \quad (1.3)$$

де

$p_t(x, y)$ – значення яскравості пікселя у t -му кадрі,

$p_{t-1}(x, y)$ – значення яскравості пікселя у $(t - 1)$ -му кадрі.

Рух пікселів вважається значним, якщо він досягає порогу T . Такий рух називається переднім планом FG_t :

$$FG_t = 1, \text{ якщо } FD_t > T$$

або

$$FG_t = 0, \text{ інакше} \quad (1.4)$$

Якщо немає великого руху, пікселі вважаються стабільними або фоновими. Кадр фонового розрізнення можна записати у вигляді:

$$BD_t = |p_t(x, y) - BM_{t-1}(x, y)|, \quad (1.5)$$

де $BM_{t-1}(x, y)$ – модель фону.

Надалі пікселі класифікуються на пікселі переднього або заднього плану за допомогою рівняння:

$$FG_t = 1, \text{ якщо } D_t(x, y) > T$$

або

$$FG_t = 0, \text{ інакше} \quad (1.6)$$

У роботі [9] авторами запропоновано розглянути алгоритм виявлення та відстеження певного об'єкту у приміщенні за допомогою малогабаритного БПЛА. Загальна схема цього методу полягає в виконанні спочатку задачі пошуку камерою дрона потрібного об'єкта за розміром і кольором. Якщо об'єкт не розпізнано, дрон розвертається і переміщується в інше місце. Запропоноване програмне рішення опрацьовує зображення у форматі RGB, який пізніше конвертується в формат HSV. На цьому етапі всі об'єкти, які не збігаються з кольором потрібного об'єкта, відкидаються. Результатом є бінарне зображення. Після чого обчислюється положення і кут нахилу об'єкта для його відстеження.

1.2.3 Метод Віоли-Джонса для виявлення об'єктів в режимі реального часу

Метод Віоли-Джонса визначається своєю ефективністю та швидкістю виявлення об'єктів на зображеннях, що отримало особливий успіх в контексті розпізнавання обличчя людини на час написання роботи. Його основу складають ознаки Хаара [10], які використовують лінійні, прямокутні фільтри для виявлення змін у яскравості на зображенні. Процес розпізнавання реалізується через застосування каскадного класифікатора над інтегральним зображенням, який поєднує ряд класифікаторів алгоритму AdaBoost, якому приділена увага п. 1.2.4, при послідовному їх застосуванні для точного визначення об'єкта.

Метод Віоли-Джонса використовує наступні принципи, що є суттєвими для організації власного програмного рішення:

1. *Створення інтегральних зображень*: вхідне зображення обробляється для створення інтегральних зображень, що дозволяє ефективно обчислювати суми значень пікселів на прямокутних областях (кумулятивна сума, яка спочатку обчислена зліва – направо, потім згори – донизу).

2. *Використання ознак Хаара*: базові ознаки Хаара представляють собою маску, яка складається з позитивних і негативних областей, та дозволяють визначати локальні особливості, такі як границі об'єктів.

3. *Адаптивний вибір ознак*: здійснюється адаптивний вибір ознак для визначення тих областей, де може знаходитися об'єкт. Кожна ознака оцінюється за здатністю ефективно розділяти позитивні та негативні області (позитивні – об'єкт присутній, негативні – об'єкт відсутній).

4. *Створення каскаду класифікаторів*: послідовність класифікаторів формує каскад, де кожен класифікатор AdaBoost використовується для відсіювання негативних областей. Каскад складається зі стеку етапів, де кожен етап має свій власний набір ознак.

5. *Каскадне визначення об'єкта*: зображення пропускається через каскад етапів, де кожен етап визначає, чи область є об'єктом чи ні. Негативні області вилучаються на ранніх етапах, тим самим значно прискорюючи процес розпізнавання.

Однією з ключових переваг методу Віоли-Джонса є його висока швидкість та здатність працювати в режимі реального часу. Завдяки використанню класифікаторів та каскадної структури, алгоритм ефективно виключає «нецікаві» області та швидко розпізнає об'єкти. Метод Віоли-Джонса знайшов широке застосування в системах відеоспостереження та розпізнавання облич у реальному часі.

1.2.4 AdaBoost для підвищення ефективності алгоритмів класифікації

Модель називається «сильною», якщо припускає невелику кількість помилок класифікацій. Модель називається «слабкою» при отриманні великої кількості помилок класифікацій та не дозволяє надійно розділити класи. Тому, визначено процедуру бустингу для усунення недоліків всіх попередніх

алгоритмів для кожного наступного, що є послідовною побудовою композиції алгоритмів машинного навчання. Одним з найбільш досконалих алгоритмів бустингу є алгоритм AdaBoost.

Алгоритм AdaBoost є комплексом методів, який сприяє підвищенню точності аналітичних моделей.

Алгоритм AdaBoost є адаптивним тому, що кожен наступний класифікатор будується з тих об'єктів, які були неправильно класифіковані минулими. Даний алгоритм є чутливим до шуму і викидів в даних та менш схильний до перенавчання.

Робота алгоритму AdaBoost на задачі побудови бінарного класифікатора наведена нижче. Початковий набір точок представлений у вигляді:

$$(x_1, z_1), \dots, (x_m, z_m) \quad (1.7)$$

де $x_i \in X, z_i \in Y = \{-1, +1\}, i = 1, 2, \dots, m, m \in N$.

Початкові ваги визначені як:

$$D_1(i) = \frac{1}{m}, \text{ де } i = 1, \dots, m; m \in N \quad (1.8)$$

Для всіх $t = 1, \dots, T$ визначимо класифікатор:

$$h_t : X \rightarrow \{-1, +1\}, \quad (1.9)$$

що мінімізує зважену похибку класифікації, яка наведено у формулі:

$$h_t = \min_{h_j \in H} \varepsilon_j, \quad (1.10)$$

де $\varepsilon_j = \sum_{i=1}^m D_t(i) [z_i \neq h_j(x_i)]$

Якщо $\varepsilon_t \geq 0.5$, то дії алгоритму припиняються.

Вибір $a_t \in R$, котрий зазвичай дорівнює $a_t = 0.5 \ln((1 - \varepsilon_t) / \varepsilon_t)$, де ε_t є зваженою похибкою класифікатора h_t .

Наступним кроком є оновлення вагів, які визначають важливість кожного з об'єктів множини навчання для класифікації:

$$D_{t+1}(i) = \frac{D_t(i)}{DZ_t} e^{-a_t z_i h_t(x_i)} \quad (1.11)$$

де DZ_t – нормалізуючий параметр, який обирається так, щоб $\sum_{i=1}^m D_{t+1}(i) =$

1

Побудова результуючого «сильного» класифікатору приведена у формулі:

$$H(x) = \text{sign} \left(\sum_{t=1}^T a_t h_t(x) \right) \quad (1.12)$$

Для подальшого дослідження у дисертаційній роботі наведений алгоритм AdaBoost може бути використаний завдяки його переваг:

- 1) доволі проста реалізація;
- 2) дуже гарна узагальнююча здатність, яка може покращуватися зі збільшенням числа базових алгоритмів;
- 3) можливість виявлення об'єктів, які є шумовими викидами;
- 4) власні обчислювальні витрати цього алгоритму невеликі.

При роботі з алгоритмом AdaBoost потрібно враховувати вимогу до досить масивних навчальних вибірок та можливість побудови неоптимальних наборів базових алгоритмів з необхідністю повторного навчання, що є недоліками та можуть вплинути результати моделювання.

Іноді відбувається перенавчання за наявності досить високого рівня шуму даних: експоненційна функція втрат (1.11) дуже збільшує ваги «найважчих» об'єктів, на яких помиляються багато «слабких» алгоритмів. Найчастіше саме ці об'єкти виявляються шумовими викидами, що змушує алгоритм AdaBoost налаштовуватися на шум і веде до перенавчання. Ця проблема вирішується шляхом видалення цих шумових викидів або застосування таких функцій втрат, які є менш «агресивними».

1.2.5 Алгоритм Fast R-CNN

Алгоритм Fast R-CNN є одним з найбільш популярних алгоритмів у сфері виявлення об'єктів. На відміну від простої класифікації зображень, яка класифікує все зображення за певною категорією, виявлення об'єктів має на меті точне визначення місцезнаходження об'єктів на зображенні, а також їх класифікацію.

У порівнянні з класифікацією зображень, виявлення об'єктів є складнішим завданням, оскільки вимагає високої точності, до того ж в реальному часі. Отже, багато з існуючих алгоритмів не задовольняють вимогам. Для вирішення цієї проблеми, були спроби використання безлічі алгоритмів, у тому числі безперервно адаптивне середнє зміщення (Camshift) з фільтром Калмана та відстеження-навчання-виявлення (TLD), відстеження на основі HOG та SIFT. Алгоритм Camshift, що більш детально розглянутий в підпункті 1.3.2, використовує статистику кольору як функцію відстеження, що робить його схильним до помилкової ідентифікації, коли цільовий колір і колір фону схожі. Алгоритм TLD є занадто повільним для досягнення мети відстеження, що призводить до низької продуктивності в реальному часі. В алгоритмі відстеження HOG + Support Vector Machine (SVM) важко отримати ефективні функції, коли об'єкт знаходиться далеко.

На відміну від перерахованих вище традиційних методів, алгоритми глибокого навчання більш надійні і точні та дозволяють швидко виявити і локалізувати об'єкт за зображенням.

Алгоритм Fast R-CNN отриманий з R-CNN, який є модифікацією алгоритму CNN побудованого для розв'язання задачі класифікації зображень. На відміну від класифікації зображень, виявлення вимагає локалізації об'єктів усередині зображення, що вперше було досягнуто за допомогою R-CNN. CNN може призвести до помітного підвищення ефективності виявлення заперечень у PASCAL VOC в порівнянні з системами, заснованими на більш простих функціях, подібних до HOG. Для виявлення об'єктів на зображеннях алгоритм R-CNN може вирішити проблему локалізації алгоритму CNN, діючи в рамках парадигми розпізнавання за областю (region based detection, RBD), яка виявилася успішною як для виявлення об'єктів, так і для семантичної сегментації. Загальна схема роботи RBD показана на рисунку 1.6.

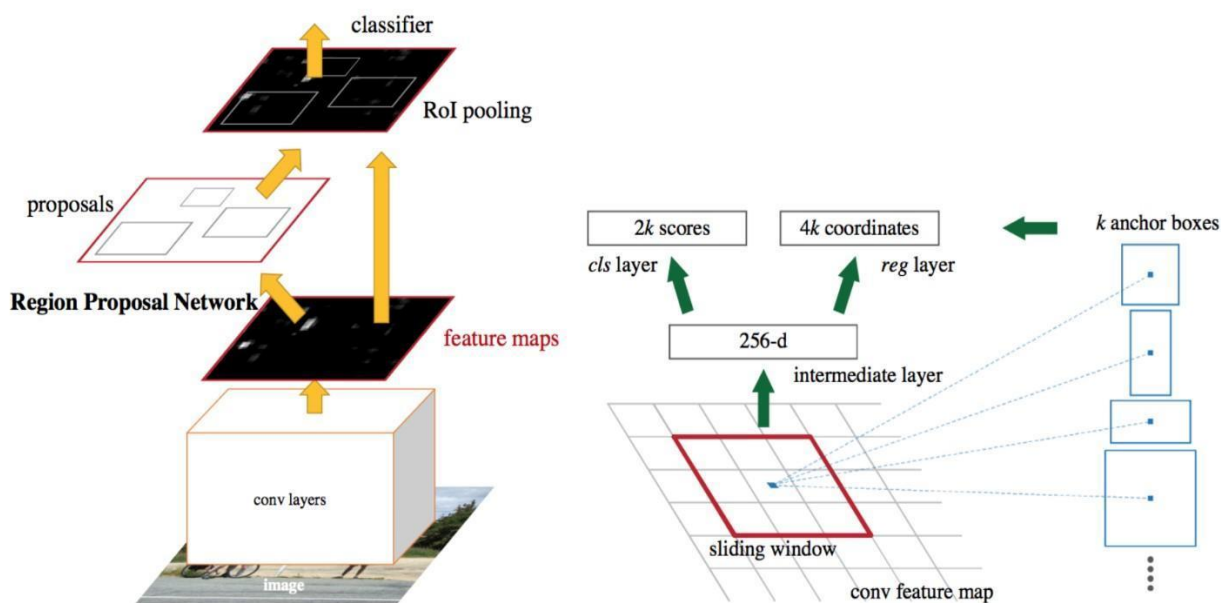


Рисунок 1.6 – Загальна схема роботи RBD моделей [10]

Під час роботи алгоритм виконує такий порядок дій:

- 1 Метод генерує близько 2000 незалежних від категорії регіональних пропозицій для вхідного зображення.
- 2 Вектор ознак фіксованої довжини витягується з кожної пропозиції із використанням CNN.
- 3 Класифікується кожен регіон з лінійними SVM для конкретної категорії.
- 4 Класифікація об'єктів та їх розташування.

Алгоритм R-CNN досяг чудової точності виявлення об'єктів, використовуючи глибоку мережу ConvNet для класифікації пропозицій об'єктів. Але основним недоліком цього алгоритму виділяють повільність роботи, оскільки він виконує прямий прохід ConvNet для кожної пропозиції об'єкта без спільного використання обчислень.

Тому було запропоновано мережі об'єднання просторових пірамід (SPPnet) для прискорення R-CNN за допомогою спільного використання обчислень. Метод SPPnet обчислює згорткову карту ознак для всього вхідного зображення, а потім класифікує кожну пропозицію об'єкта, використовуючи

вектор ознак, витягнутий із загальної карти об'єктів. Крім того, R-CNN вимагає фіксованого розміру вхідного зображення, що обмежує як співвідношення сторін, так і масштаб зображення. Поточний метод дозволяє змінити розмір зображення до фіксованого розміру, в основному шляхом обрізання або деформації, що призведе до небажаних спотворень або втрати інформації, у той час як SPPnet може об'єднувати зображення будь-якого розміру для отримання зображення сталої розмірності. На практиці SPPnet прискорює роботу R-CNN у 10–100 разів. Однак, метод SPPnet має недолік: навчання являє собою багатоетапний конвеєр (як і в R-CNN), який включає вилучення ознак, точне налаштування мережі з втратами, навчання SVM і підбір регресорів з обмеженою зоною. Більш того, алгоритм тонкого налаштування не може оновлювати згорткові шари, які передують об'єднанню просторових пірамід. Це обмеження обмежує точність дуже глибоких мереж.

Надалі був запропонований удосконалений алгоритм, який усуває недоліки R-CNN і SPPnet, названий Fast R-CNN з підвищеною швидкістю і точністю. Fast R-CNN може оновлювати всі шари мережі на етапі навчання, не потребує дискового сховища для кешування ознак та ознаки зберігаються у відеопам'яті.

На рисунку 1.7 показано архітектуру Fast R-CNN [11]. Мережа Fast R-CNN використовує, як вхідні дані всі зображення та набір пропозицій об'єктів:

- 1 Все зображення з 5 згортковими шарами та максимальним пулом шарів обробляється для створення згорткової карти ознак.

- 2 Для кожної пропозиції об'єкта шар об'єднання ROI отримує вектор ознак фіксованої довжини з карти ознак.

- 3 Кожен вектор ознак подається на послідовність повністю з'єднаних шарів, які розгалужуються на 2 вихідні шари:

- один шар, який виробляє оцінки ймовірності softmax для K класів об'єктів плюс "фон";

– другий шар, який виводить 4 дійсних числа для кожного з K класів об'єктів; кожен набір з 4 значень кодує уточнені позиції обмежувальних рамок для одного з K класів.

Fast R-CNN досягає швидкості, наближеної до реального часу, використовуючи дуже глибокі мережі, ігноруючи час, що витрачається на пропозиції регіонів. Пропозиції – це вузьке місце в обчисленнях найсучасніших систем виявлення.

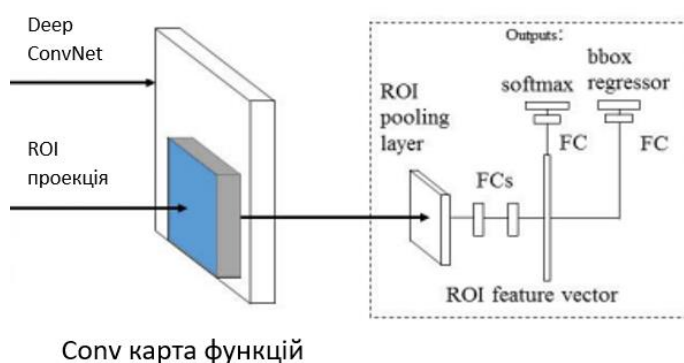


Рисунок 1.7 – Огляд системи Fast R-CNN

Метод SPPnet та алгоритм Fast R-CNN зменшують час роботи цих мереж виявлення, що робить обчислення пропозицій регіонів вузьким місцем.

Для вирішення цієї проблеми було запропоновано новий метод – мережу пропозицій регіонів (RPN). Метод RPN поділяє функції згортки повного зображення з мережею розпізнавання, що дозволяє майже безкоштовно обчислювати пропозицію регіону. Це повністю згорнута мережа, яка одночасно прогнозує межі об'єктів і оцінки об'єктів у кожній позиції. RPN навчаються наскрізно, щоб генерувати високоякісні пропозиції регіонів, які використовуються Fast R-CNN для виявлення. Таким чином, Faster R-CNN можна розглядати як комбінацію RPN і Fast R-CNN. Вибірковий пошук був замінений на RPN для генерації пропозицій. Faster R-CNN [12] об'єднує пропозицію регіону, вилучення ознак, класифікацію та прямокутне уточнення в цілу мережу, таким чином досягаючи виявлення в реальному часі.

1.2.6 Модель Faster R-CNN

До групи розпізнавання за областю (region based detection, RBD), що розпізнає об'єкти у два етапи (рисунк 1.7) відноситься модель Faster R-CNN. На кожному зображенні генеруються області інтересу (region of interest, ROI) і класифікуються в мережі глибокого навчання. Алгоритми RBD використовують метод пропозиції областей (region propose) для створення областей інтересу ROI для виявлення об'єктів. Попередні методи виявлення об'єктів вивчають різні завдання, такі як локалізація, класифікація та прогнозування обмежувальних рамок, використовуючи об'єднану мережу CNN. Пізніше, RCNN демонструє значне покращення в задачі розпізнавання об'єктів. RCNN обчислює місцезнаходження об'єктів та виокремлює їх, згодом кожен об'єкт класифікується за допомогою мережі глибокого навчання. В роботі [13] для збільшення швидкості навчання пропонується модель Faster RCNN.

Faster R-CNN складається з мережі вилучення ознак, яка зазвичай є попередньо навченою CNN. Наявні ще дві підмережі, які навчаються відповідно до поставленої задачі. Перша – це мережа пропозицій області (RPN, Region Proposal Network), яка, як випливає з назви, використовується для генерування пропозицій об'єктів, а друга – для попереднього визначення фактичного класу об'єкта. Основна відмінність Faster R-CNN полягає в тому, що RPN додається після останнього шару згортки. Цей механізм допомагає створювати пропозиції області без вибіркового підходу до пошуку. Після цього кроку послідовно застосовуються етапи об'єднання ROI, класифікація і регресія обмежувальної рамки. Faster R-CNN працює краще порівняно з іншими детекторами на основі RBD і має наступні переваги:

- якість виявлення за метрикою mAP вища, ніж у RCNN;
- класифікація та регресійне навчання виконується за один етап;
- для вилучення ознак потрібно менше пам'яті;

– мережа пропозиції області розраховує ROI досить швидко, що дозволяє використання моделі у реальному часі.

Важливим завданням комп'ютерного зору є розпізнавання об'єктів у різних масштабах.

Мережа піраміди ознак (Feature Pyramid Network, FPN) здатна генерувати багатомасштабне представлення ознак з високою роздільною здатністю, яке є семантично сильним. У роботі [14] запропонували фреймворк на основі вбудованої системи під назвою Deep Drone. Вона використовується як для розпізнавання, так і для відстеження. Запропонована система реалізована як на незалежному, так і на вбудованому графічному процесорі й показує стабільно швидкі і точні результати.

На рисунку 1.8 показано прогрес від R-CNN до Fast R-CNN і до Faster R-CNN.



Рисунок 1.8 – Різновид алгоритмів R-CNN, Fast R-CNN, Faster R-CNN

У сфері виявлення об'єктів відбулася зміна парадигми з появою технології YOLO. Унікальна пропозиція YOLO, розроблена Джозефом Редмоном та його командою, полягала в здатності виявляти та класифікувати об'єкти на зображеннях за один прямий прохід нейронної мережі. Цей відхід від традиційних двоетапних процесів, таких як у методах на основі R-CNN, дозволив YOLO досягти швидкості виявлення в реальному часі без суттєвого погіршення точності. YOLOv3, вдосконалена ітерація серії YOLO, принесла

низку вдосконалень. Однією з його помітних особливостей стало використання трьох різних розмірів анкерних коробок для кожної шкали виявлення. Такий підхід дозволив YOLOv3 краще виявляти об'єкти різних розмірів. Крім того, він використовував три різні масштаби виявлення, використовуючи особливості різних рівнів мережі. Ця стратегія багатомасштабного виявлення підвищила точність виявлення об'єктів різного розміру. Ще одним стрибком у YOLOv3 стало використання трьох гілок прогнозування, кожна з яких відповідає за прогнозування обмежувальних рамок, оцінок об'єктності та оцінок класів.

Розвиваючись далі, YOLOv5 з'явився не як прямий спадкоємець оригінальних розробників YOLO, а як результат еволюції, керованої спільнотою. Хоча його назва може свідчити про протилежне, YOLOv5 немає аналога v4 з тієї ж лінії. YOLOv5, порівняно з YOLOv3, може похвалитися поліпшеннями як у швидкості, так і в точності. Завдяки архітектурним змінам, вдосконаленим методам навчання та оптимізації, він забезпечує кращі показники продуктивності на різних бенчмарках.

1.2.7 Опис методу SSD

Алгоритми розпізнавання об'єктів, такі як RBD, досягли значних успіхів у точності, однак, швидкість роботи є неефективною. Методи розпізнавання з парадигми розпізнавання за один прохід (Single shot detection, SSD) мають високу швидкість і менші вимоги до пам'яті порівняно з підходами, заснованими на RBD. Алгоритми, засновані на SSD, потребують лише одного проходу, щоб розпізнати багато об'єктів, використовуючи мультибокс (відповідь мережі представлена набором векторів, які одночасно містять інформацію й про обмежувальну рамку й про клас об'єкту) на зображенні. Він значно швидший за швидкістю і має високу точність, оскільки уникає

пропозицій обмежувальних рамок (bounding box proposals), подібних до тих, що використовуються в RCNN.

В роботі [15] запропонували модель на основі глибокого навчання для розпізнавання об'єктів на зображеннях з БпЛА. Parrot AR Drone 2 використовувався для зйомки зображень, а аналіз зображень обчислюється на сервері. Дрон і сервер з'єднані через WiFi. Автори застосували модель SSD (Single Shot Detector) для розпізнавання об'єктів, оскільки вважають, що алгоритми, засновані на RBD, є ресурсозатратними. Вихідні дані SSD подаються на PID-контролер для відстеження об'єкта. PID-контролер розглядає тривимірну простір, який забезпечує кращу точність і менший час обчислень, що робить його придатним для застосувань у реальному часі.

Модель SSD представляє собою детектор одного кадру, тобто вся обробка зображення відбувається за один прохід нейронної мережі VGG16.

На рисунку 1.9 зображено схема оброблення зображення методом SSD, де 3-5 кроки це індивідуальні шари згортки, а шостий крок – фільтр прогнозування.



Рисунок 1.9 – Схема оброблення зображення методом SSD

Даний підхід дискретизує вихідний простір обмежувальних рамок в набір полів за замовчуванням для різних пропорцій і шкал на кожне розташування карти. В процесі прогнозування нейронна мережа генерує певну кількість балів для кожної визначеної категорії об'єктів в кожному визначеному об'єкті за замовчуванням і видає коригування в поле, для того щоб краще відповідати власне формі об'єкта. Також мережа об'єднує прогнози

з різних карт функцій з різними дозволами для природного управління об'єктами різних розмірів. Виявлення об'єкта SSD складається з двох частин: – вилучення картки функцій; – застосування фільтрів згортки для виявлення об'єктів.

SSD не використовує мережу пропозицій для делегованих регіонів. Замість цього він використовує дуже простий метод. Він обчислює як місце розташування, так і оцінку класів з використанням невеликих фільтрів згортки. Після вилучення карт функцій SSD застосовує 3×3 фільтри згортки до кожної комірки для прогнозування (ці фільтри обчислюють результати так само, як звичайні фільтри CNN). Кожен фільтр виводить 24 канали: 21 бал для кожного класу плюс один граничний блок.

1.2.8 Алгоритм YOLO

Алгоритм YOLO займає важливе місце у сфері виявлення об'єктів. Його ціль спрямована на збільшенні швидкості без суттєвого зниження точності. Його наступні версії, включаючи YOLOv3 і YOLOv5, ще більше закріпили його популярність в секторі виявлення в реальному часі. В той час як такі моделі, як Faster R-CNN, зосереджувалися на точності з відносно більшими обчислювальними накладними витратами, варіанти YOLO розширили межі досяжного в реальному часі, хоча іноді і ціною втрати точності.

Алгоритм YOLO представляю собою сучасну технологію розпізнавання об'єктів, яка заснована на парадигмі SSD, що використовується в сфері БпЛА для розпізнавання в реальному часі [16]. Першу модель YOLO було опубліковано в 2015 році, за цього часу модель отримала потужний розвиток, адже мала великий еволюційний потенціал. Еволюція алгоритму YOLO зображено на рисунку 1.10 [17].

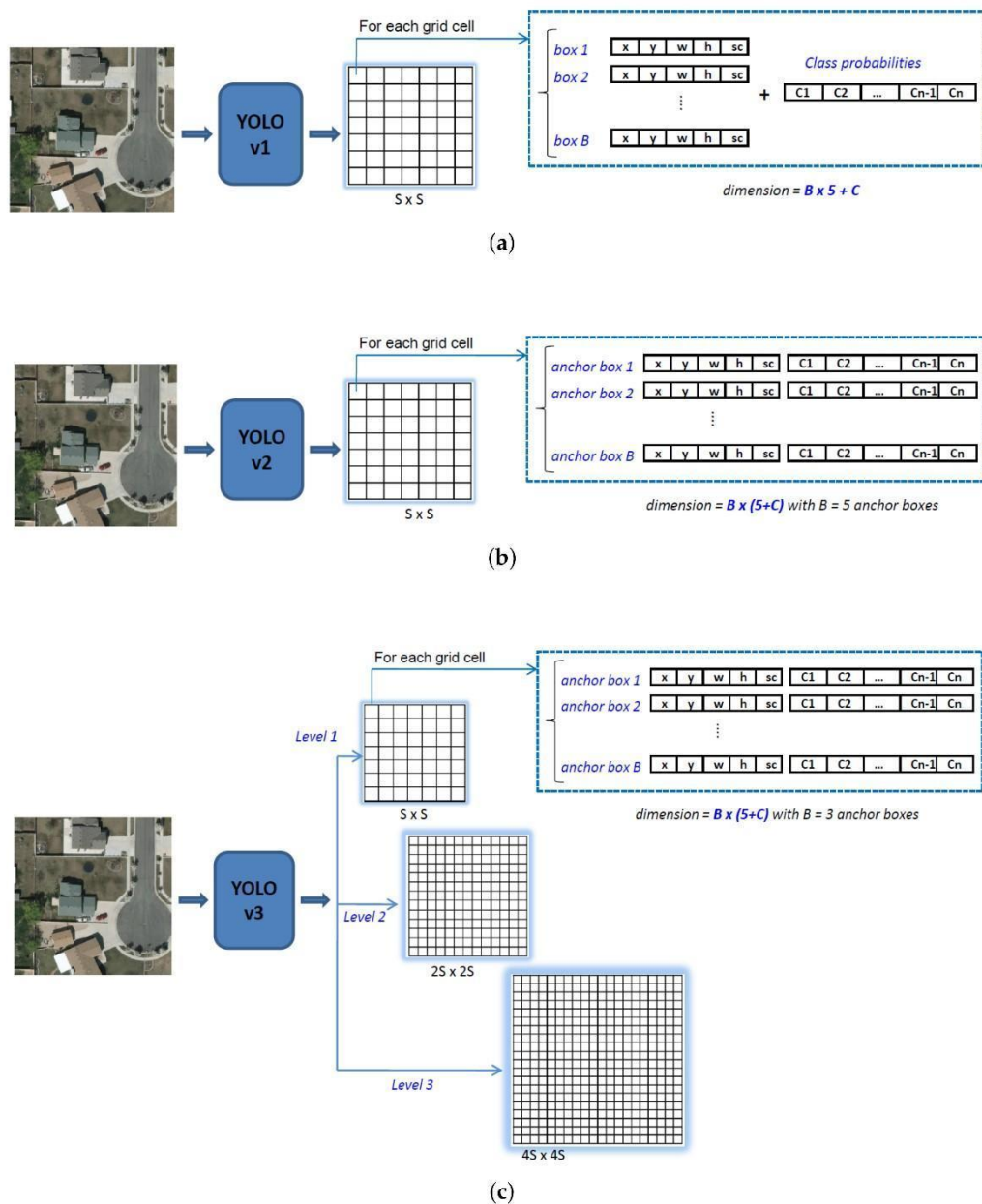


Рисунок 1.10 – Еволюція алгоритму YOLO [17]

Серед еволюційного ряду моделей сімейства YOLO спільною рисою лише залишився загальний алгоритмічний підхід. Основними чинниками еволюції алгоритму стали наступні:

- використання якорних обмежувальних рамок, що обмежувало локалізацію об'єктів рамками певної форми;
- зміна архітектури згорткової нейронної мережі, яка лежить в основі моделі YOLO;

- зміна кількості блоків, на які розбивається зображення у алгоритмі;
- використання мультимасштабного представлення для якісної роботи з об'єктами різного розміру.

З появою таких багатих наборів даних у сфері детектування об'єктів як Kitti, з'явився сплеск інновацій і вдосконалень, що охоплюють як реальні зображення, так і більш спеціалізовані сценарії.

Наприклад, Чуаньян Лю [18] запропонував алгоритм MTI-YOLO для виявлення таких цілей, як ізолятори під час інспекції ліній електропередач за допомогою БПЛА.

На основі YOLOv3-Tiny, MTI-YOLO розширює структуру, додаючи функціональну структуру злиття і модулі SPP. Він також додає вихідні шари до основи. Вдосконалення структури цього методу є відносно надлишковим, і конструкція мережі потребує оптимізації. Ойку Сахін [19] проаналізував складні проблеми в аерофотознімках БПЛА. В його роботі вихідний шар основи YOLOv3 для виявлення об'єктів різного масштабу на зображенні був розширений, збільшивши число початкових шарів виявлення з трьох до п'яти. Така структура відіграє певну роль у частині реалізації, що відповідає за злиття ознак. Однак це призводить до надто великих і складних моделей виявлення, що збільшує вартість навчання та обчислень.

Джунос разом з Дахарі [20] в своїй роботі розглядали використання БПЛА для виявлення плодів олійної пальми. Для цього вони створили масивний набір аерофотознімків з БПЛА. На основі YOLOv3-Tiny вони запропонували метод виявлення плодів олійної пальми. Метод використовує щільно зв'язну нейронну мережу і функції активації Swish, а також додає новий шар виявлення. Функція активації, обрана цією моделлю, схильна до погіршення продуктивності в глибоких мережах, а доданий шар функцій, безсумнівно, сповільнює швидкість виявлення моделі.

Цзя Гуо та ін. [21] запропонували вдосконалений метод виявлення YOLOv4 для малих цілей, таких як антивібраційні молотки в лініях

електропередач на аерознімках БпЛА. Для покращення здатності мережі витягувати ознаки, метод додає модулі ReceptiveField Block (RFB) в структуру. У запропонованому методі відсутня дискусія щодо місця додавання модулів, а вдосконалена стратегія є відносно простою.

Янбо Ченг [22] запропонував вдосконалений метод YOLO для розмитого зображення, спричиненого тремтінням камери під час аерофотозйомки з БпЛА, експозицією, спричиненою нерівномірним освітленням, та шумом під час передачі. Цей метод використовує різноманітні методи покращення даних, такі як афінне перетворення, обробка розмиття за Гаусом та перетворення градацій сірого для посилення можливостей попередньої обробки даних у YOLOv4, що використовується для полегшення проблеми складного навчання через малу кількість даних. Недоліком цього методу є те, що йому не вистачає цілеспрямованих модифікацій структури самої моделі.

На основі YOLOv5 Вей Дінг та Лі Джанг [23] додали модуль згортки блоків уваги (Convolutional Block Attention Module, CBAM), щоб розрізнити будівлі різної висоти на аерознімках з БпЛА. Основа вдосконаленої моделі покращує можливості вилучення ознак, але слід зазначити, що обсяг обчислень збільшиться через додавання інших модулів.

Сюй Вень Ванг з командою [24] запропонували метод виявлення LDS-YOLO який враховує характеристики малих цілей і незначні деталі мертвих дерев на аерофотознімках БпЛА, який потім було вдосконалено на основі YOLOv5. В роботі було побудовано новий модуль виділення ознак, введено метод SoftPool в модуль SPP, а традиційні згортки замінено на згортки, що розділяються за глибиною. Цей метод дає хорошу продуктивність. Хоча згортка, що розділяється за глибиною, зменшує параметри моделі, але в процесі навчання легше не навчитися розпізнавати цільові ознаки через недостатню кількість вибірок.

Для вирішення проблеми низької ефективності виявлення пошкоджених доріг на аерознімках БпЛА Ю Чен Лю та ін. [25] представили метод виявлення M-YOLO. Цей метод замінює основу YOLOv5s на MobileNetv3 і вводить мережеву структуру SPPNetwork, що сприяє підвищенню швидкості виявлення моделі. Слід зазначити, що збільшення швидкості часто супроводжується погіршенням точності виявлення.

На основі YOLOv5s Руй Чжан і Чуанбо Вен [26] запропонували метод виявлення дефектів лопатей вітрових турбін на аерознімках БпЛА, названий SOD-YOLO. SOD-YOLO додає шар виявлення дрібних об'єктів, використовує алгоритм К-середніх для кластеризації та отримання якірних блоків і додає модулі CBAM до системи. Крім того, використання алгоритму обрізання каналів зменшує обчислювальну вартість моделі, одночасно збільшуючи швидкість виявлення. Однак, цей метод не вирішує проблему, пов'язану з тим, що початкові опорні точки, як правило, є локально оптимальними рішеннями через кластеризацію за методом К-середніх.

Підсумовуючи, можна сказати, що при вдосконаленні моделі важливо збалансувати співвідношення між точністю виявлення та швидкістю обчислень. Хороший метод виявлення повинен намагатися враховувати два вищезгадані моменти. Найпопулярнішим методом виявлення серії YOLO є YOLOv5, який базується на YOLOv4 і має чотири версії: s, m, l і x. YOLOv5x має великий розмір і складний в обчислювальному плані. YOLOv5s та YOLOv5m швидші, але недостатньо точні. YOLOv5l добре працює з точки зору швидкості і точності і подібний до YOLOv4 з точки зору загальних параметрів і загальної кількості операцій з плаваючою комою в секунду (FLOPS).

Структуру YOLOv5l можна розділити на три частини, а саме: кістяк, шию та голову (рисунок 1.11). Кістяк також відомий як мережа вилучення ознак. Коли зображення подається на вхід, вилучення ознак виконується кістяком. Вхідне зображення спочатку проходить через модуль Focus. Цей

модуль отримує відповідне власне значення для кожної іншої точки і об'єднує чотири незалежні шари ознак для отримання остаточного результату. У цей час інформація про ширину та висоту зображення буде сконцентрована в каналі, що вирішує проблему втрати інформації, спричинену заниженою дискретизацією.

YOLOv5 використовує функцію активації SiLU, яку можна розглядати як згладжену функцію активації ReLU. SiLU не має верхньої межі, але має нижню межу і є немонотонною. Вона все ще зберігає хорошу продуктивність на глибоких мережах, і це корисно для моделі, щоб покращити ефект фільтрації при збільшенні глибини.

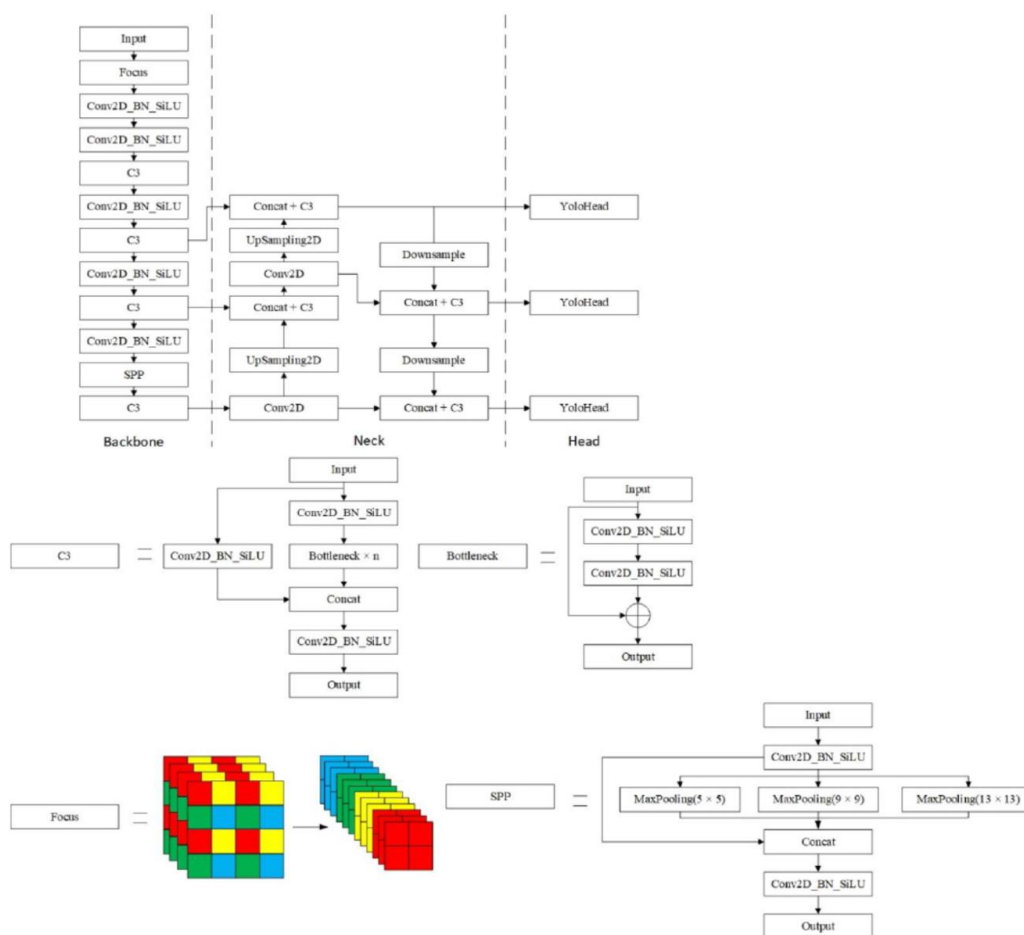


Рисунок 1.11 – Структура YOLOv5 [26]

Ядро YOLOv5 містить модуль SPP, який виконує екстракцію ознак шляхом максимального об'єднання ядер різного розміру, розширюючи

сприйнятливим полем моделі. Для злиття інформації про ознаки в різних масштабах шийна частина використовує три мапи ознак різного розміру, вилучені ядром для злиття ознак. У цій частині все ще використовується структура PANet, яка, на основі FPN, додає канал з мілкою мережею до глибокої мережі. Це допомагає об'єднати місцезнаходження та семантичну інформацію поверхневих та глибинних об'єктів, тим самим покращуючи використання інформації та прискорюючи ефективність її поширення. Голову моделі можна розглядати як класифікатор і регресор YOLOv5. За допомогою згортки 1×1 визначається, чи існує об'єкт на карті ознак, що йому відповідає. Під час навчання використовується мозаїчне доповнення даних, яке збагачує фон виявлених об'єктів і допомагає підвищити ефективність пакетної нормалізації.

Використовується згладжування міток, оскільки воно допомагає зменшити ризик перебору моделі та покращує узагальнення. Використовується адаптивний підхід до якірних блоків, який полегшує автоматичне встановлення початкового розміру якірного блоку при зміні різних наборів даних. Структура YOLOv5 показана на рисунку 2.

Наразі різних модифікацій моделі YOLO опубліковано досить багато, серед них можна відокремити основне еволюційне дерево, самою новою моделлю якого є YOLOv8. Модель YOLOv8 є найдосконалішою для задачі розпізнавання об'єктів у балансі точність-швидкодія [27], на момент написання цієї роботи.

1.2.9 Бібліотека OpenCV

Значний внесок у вдосконалення алгоритмів машинного зору здійснив проєкт OpenCV. Він був запущений з ініціативи компанії Intel у 1999 році. OpenCV – це opensource-бібліотека комп'ютерного зору, обробки зображень і чисельних алгоритмів з досить широким функціоналом. Реалізована бібліотека на C/C++, є підтримка Python, Java, Ruby, Matlab, Lua та інших мов.

Зараз проєкт OpenCV підтримується некомерційною організацією OpenCV.org.

Можливості OpenCV включають:

- кадрування;
- зміна розміру зображення, що потрібно для згорткових нейронних мереж, щоб зменшити навантаження на систему;
- поворот зображення;
- переведення кольорового зображення в чорно-біле/градації сірого;
- методи розмиття/ згладжування;
- наявність інструменту рисування прямокутників і ліній, якими позначаються межі об'єктів при розмітці датасета для систем розпізнавання;
- наявність інструменту створення тексту на зображенні, що дозволяє використовувати цю функцію при роботі з відео для динамічного показу певних параметрів частоти кадрів відеопотоку, кількості певних об'єктів, що відслідковуються в реальному часі в системах, додаткова інформація при налаштуванні.;
- наявність технології розпізнавання обличчя, яка реалізована завдяки попередньо натренованим моделям, що засновані на нейромережах.

Бібліотека містить понад 2500 оптимізованих алгоритмів, що включає в себе вичерпний набір як класичних, так і сучасних алгоритмів комп'ютерного зору та машинного навчання. Бібліотеку у своїх продуктах використовують такі відомі компанії, як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, а також безліч стартапів.

1.3 Аналіз існуючих рішень задачі відстеження об'єктів

Відстеження об'єктів є терміном, який використовується для задачі визначення положення рухомих об'єктів, а також для відстеження їх з відеопослідовностей. Класифікація методів і задач відстеження об'єктів дуже

різноманітна через великий простір варіацій постановки проблеми відстеження та підходів до їх вирішення.

Методи відстеження об'єктів поділяються в основному за кількістю об'єктів відстеження: для одиничного об'єкту та для багатьох об'єктів. Зазвичай, відстеження одиничного об'єкту (single object tracking, SOT) потребує не дуже складного алгоритму та може проводитися без навчання (налаштування забезпечують параметри), але значно можуть бути обмежені в автономності. Таким чином, для використання алгоритмів відстеження одиничного об'єкту необхідно виокремити цей об'єкт певним чином, найчастіше використовується мануальне виділення, або стаціонарне (вважається, що об'єкт початково знаходиться в певній частині зображення) на початку процесу відстеження.

Відстеження багатьох об'єктів (multiple object tracking, MOT) має однозначну перевагу в автономності, адже не потребує початкове виділення об'єктів мануально чи стаціонарно. Натомість, алгоритми відстеження багатьох об'єктів більш комплексні та потребують навчання, або вже навчену модель розпізнавання, що відповідатиме задачі відстеження. Також визначним є принцип роботи алгоритму для відстеження багатьох об'єктів, для даної роботи найбільше підходить парадигма відстеження через розпізнавання (tracking by detection, TBD або detection based tracking). Парадигма відстеження через розпізнавання означає наявність натренованої моделі розпізнавання об'єктів та алгоритму для їх відстеження (рисунок 1.12) [28].

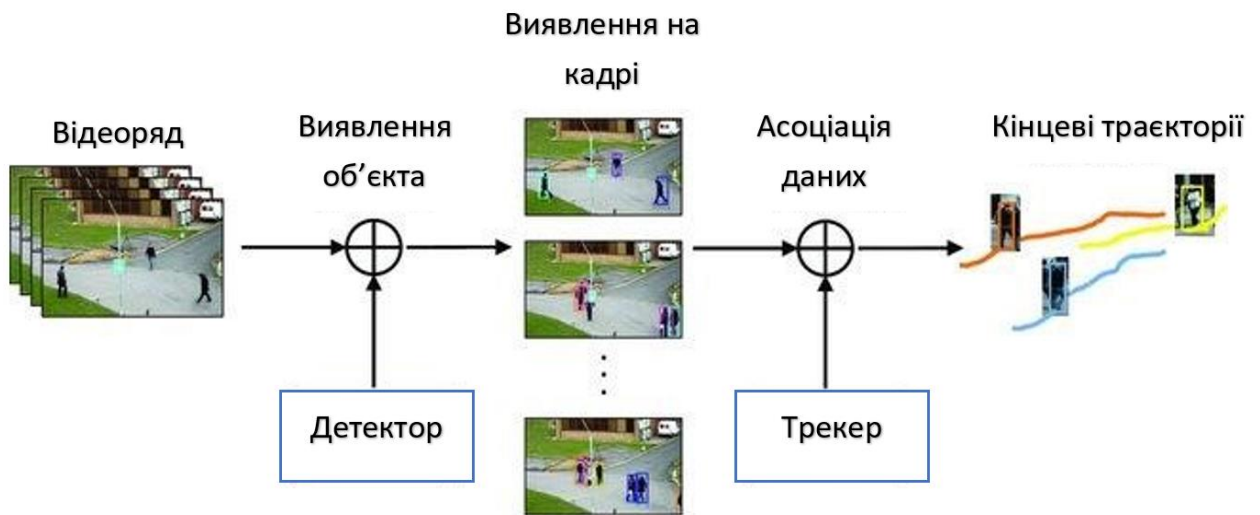


Рисунок 1.12 – Схема роботи алгоритму відстеження через розпізнавання

Також методи відстеження об'єктів поділяються на ті, що здатні працювати в режимі реального часу (online), та ті, що мають частоту відстеження нижче за необхідну частоту зміни кадрів (offline).

1.3.1 Задача відстеження об'єктів

Проблему відстеження багатьох об'єктів найчастіше представляють як графову модель або регресійну модель передбачення багатьох змінних. Для регресійної моделі задача відстеження зводиться до максимізації функції правдоподібності щодо розпізнаних на поточному кроці об'єктів до відстежених на попередніх кроках.

Для задачі відстеження багатьох об'єктів через розпізнавання у графовому вигляді (рисунок 1.13) задано множину вершин графу, які визначають розпізнані об'єкти на кожному кроці t :

$$O^t = \{O_1^t, O_2^t, \dots, O_n^t\} \quad (1.13)$$

Ребрами в такому представленні є міра подібності $r_{i,j}$ між вершинами-об'єктами O_1^t та O_1^{t-1} з попереднього кроку.

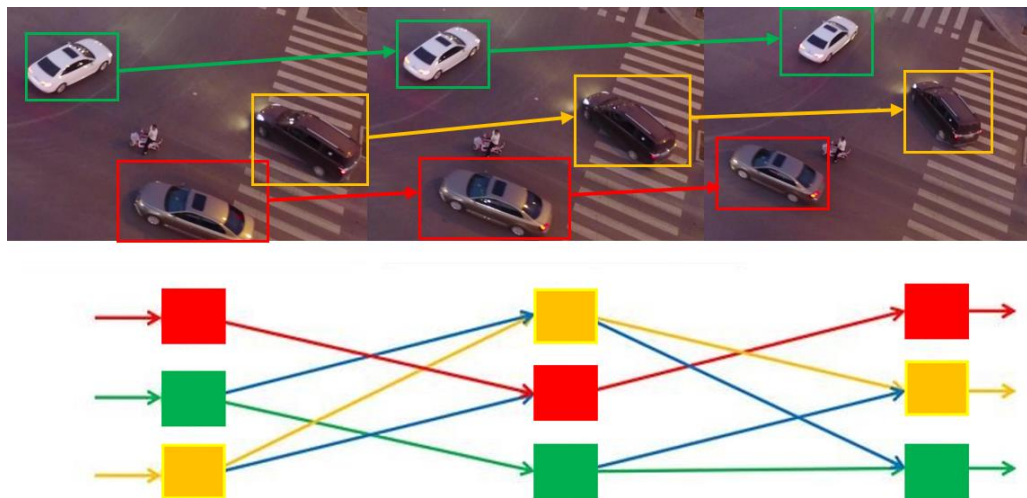


Рисунок 1.13 – Графове представлення задачі відстеження багатьох об'єктів через розпізнавання

Тоді задача відстеження зводиться до задачі про призначення між вершинами O_1^t та O_1^{t-1} , де необхідно максимізувати загальну суму мір подібності:

$$\sum_{O_i^t \in O^t} r_{i,j} \quad (1.14)$$

1.3.2 Безперервне адаптивне відстеження середнього зсуву

Безперервне адаптивне відстеження середнього зсуву (Continuously adaptive mean shift tracking, CAMShift) є методом відстеження об'єктів на кадрах відеоряду на основі кольорів. Цей метод зосереджений на методі відстеження середнього зсуву (mean-shift tracking method) і спочатку був запропонований для моніторингу людських обличчя в рамках користувацького інтерфейсу. Перевага цього підходу полягає в тому, що він дає змогу налаштовувати вікно пошуку. Метод середнього зсуву є покроковою технікою, яка вибирає вікно пошуку і надає історію положення, типу, форми і розміру об'єкта.

Алгоритм для методу CAMShift включає наступні кроки:

- 1 Вибір початкового положення вікна пошуку.
- 2 Виконання зміщення за середнім значенням.
- 3 Обчислення середньої позиції у вікні пошуку.
- 4 Вікно пошуку розміщується на основі середнього положення, яке визначається на попередньому кроці.
- 5 Повторення пунктів 3 та 4 до досягнення збіжності процесу.
- 6 Встановлення розміру вікна пошуку, який дорівнює функції нульового моменту з кроку 2.

Розмір вікна пошуку обчислюється за формулою 1.15.

$$S = 2 * \sqrt{\frac{m_{00}}{256}} \quad (1.15)$$

де m_{00} – це нульовий момент.

В науковій публікації [29] запропоновано модель розпізнавання та відстеження об'єктів для спостереження цілей. Модель спрямована на виявлення незвичайних подій на відео з БПЛА. Відеоряд знімається за допомогою БПЛА Phantom та розпізнавання об'єктів відбувається на основі методу адаптивного віднімання фону. Відстеження об'єктів здійснюється за допомогою методів Lucas Kanade та CAMShift.

1.3.3 Опис TransMOT

Досягнення в галузі глибинного навчання привели до вивчення просторово-часових зв'язків за допомогою глибинного навчання. Зокрема, успіх трансформерів (transformer) запропонував нову парадигму моделювання часових залежностей за допомогою потужного механізму самоуваги (selfattention mechanism). Але методи відстеження об'єктів на основі трансформерів певний час не могли досягти конкурентної продуктивності відносно інших методів з причин того, що:

- відео може містити велику кількість об'єктів і моделювання просторово-часових зв'язків цих об'єктів за допомогою загального трансформера є неефективним;
- для моделювання довготривалих часових залежностей трансформер потребує багато обчислювальних ресурсів та даних;
- використання недосконалих методів розпізнавання об'єктів.

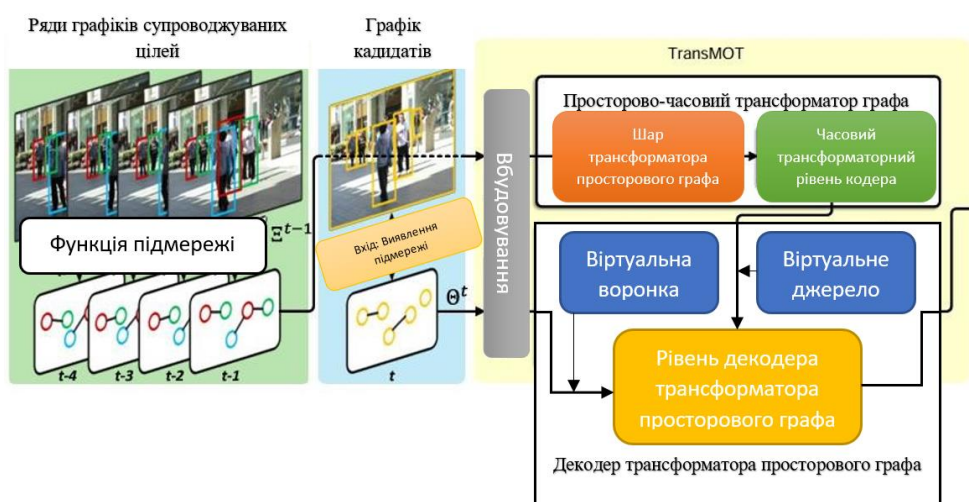


Рисунок 1.14 – Схема роботи методу відстеження TransMOT

Інноваційний просторово-часовий граф-трансформер TransMOT запропоновано у роботі [30], який вирішує вище перелічені проблеми. Модель-трансформер впорядковує траєкторії всіх відстежуваних об'єктів у вигляді серії розріджених зважених графів, які будуються з використанням просторових взаємозв'язків цілей. Потім TransMOT використовує ці графи для створення шару кодера просторового графового трансформера, шару кодера часового трансформера і шару декодера просторового трансформатора для моделювання просторово-часових зв'язків об'єктів (рисунок 1.14). Розрідженість представлень зважених графів робить їх більш ефективними в обчислювальному плані під час навчання та виведення, оскільки модель використовує структуру об'єктів.

1.3.4 Метод відстеження BYTE Track

У роботі [31] було представлено метод відстеження багатьох об'єктів в режимі реального часу BYTE Track з інноваційним алгоритмом асоціації BYTE. Метод відстеження BYTE Track вирізняється високою швидкістю, точністю та найвищою якістю відстеження за метрикою MOTA у виклику MOT17 (Multi-Object Tracking) (рисунок 1.15). Його основною перевагою є здатність ефективно відстежувати об'єкти, з малою оцінкою впевненості (confidence score).

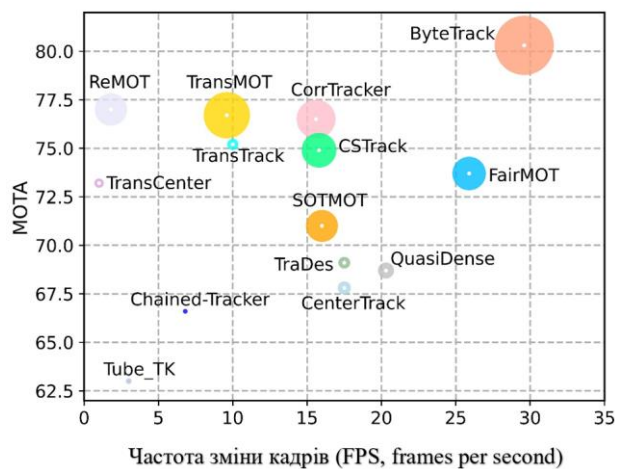


Рисунок 1.15 – Результати виклику MOT17 за метрикою MOTA та частотою кадрів у FPS

Метод BYTE Track належить до типу методів відстеження через розпізнавання, тому потребує моделі розпізнавання багатьох об'єктів (у вищезгаданій роботі [31] було використано модель YOLO). На вхід BYTE Track надходять розпізнані об'єкти у вигляді множини обмежувальних рамок разом з оцінкою впевненості для кожного об'єкту. Алгоритм заснований на принципі розбиття розпізнаних на поточному кадрі об'єктів на групи з високим та низьким оцінками впевненості, що визначає параметр порогу. Проводиться операція асоціації для об'єктів з високою оцінкою впевненості: у випадку невдачі дані об'єкти сприймаються як нові та додаються до

відстежуваних. Для об'єктів з низькою оцінкою проводиться також процес асоціації, але у випадку неспівпадіння дані об'єкти відкидаються.

Алгоритм BYTE Track використовує фільтри Калмана для передбачення позиції об'єкту на наступному кадрі відеоряду при врахуванні нормального відхилення. Детальний розгляд фільтру Калмана наведено в пункті 2.4.2. При роботі алгоритму кожному відстежуваному об'єкту призначається відповідний фільтр Калмана, який передбачає його місцезнаходження на наступному кадрі та оновлюється згідно з реальним місцезнаходженням на поточному кадрі після етапу асоціації. Для асоціації об'єктів використовується метрика IOU (Interception over union, перетин над об'єднанням), що перетворює задачу асоціації на стандартну задачу про призначення, для вирішення якої використовується Угорський метод (детальний опис угорського методу описан в пункті 2.4.2).

1.4 Висновки до розділу 1

В першому розділі було розглянуто спеціалізовану літературу та інформаційні технології для визначення актуальності теми дослідження даної дисертаційної роботи. Розглянуто існуючі методи та рішення для виконання поставленої задачі.

Наведено статистичні дані та висвітлено проблематику задач комп'ютерного зору в сфері БпЛА з описом їх причин на практиці.

Поставлено формальну задачу для виявлення та розпізнавання об'єктів на зображеннях та розглянуто методи та алгоритми як традиційні (віднімання фону, метод Віоли Джонса), так і засновані на глибокому навчанні (мережі Faster R-CNN, SSD, YOLO).

При врахуванні проведеного аналізу переваг та недоліків кожного з описаних методів, для розроблення технології за базову концепцію обрано архітектурне рішення інноваційної мережі YOLO v9 з декотрими

запропонованими методами виявлення та розпізнавання. Даний вибір ґрунтується на найкращому балансі точності розпізнавання до швидкості роботи, що дає можливість реалізувати розпізнавання об'єктів в реальному часі.

Для задачі відстеження об'єктів розглянуто класифікацію напрямку специфік багатьох підзадач та визначено, що найбільше відповідає темі дослідження методи відстеження багатьох об'єктів через розпізнавання в реальному часі. Оглянуто різні методи відстеження: як і для одного об'єкту (з можливою модифікацією для багатьох) так і для багатьох. Обрано метод BYTE Track, як найшвидший для відстеження багатьох об'єктів і з достатньою точністю ідентифікації, який заснований на фільтрах Калмана та має перевагу при недостатній оцінці впевненості в наявності об'єкта.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ОБЧИСЛЮВАЛЬНИХ АСПЕКТІВ МЕТОДІВ ВИЯВЛЕННЯ, РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ

Широкої актуальності набуває ідея створення програмних інструментів, що дозволяють організувати діяльність з розроблення методів та алгоритмів машинного зору для навігації БпЛА, забезпечити можливість отримання тестових відеоданих та проведення експериментальних досліджень з об'єктивною кількісною оцінкою ефективності підходів, що реалізуються.

Застосування технологій комп'ютерного зору для навігації літальних апаратів в основному зводиться до того, що необхідно в той чи інший спосіб зіставити дані поточної відеозйомки з отриманими наперед еталонними зображеннями місцевості. Через велику кількість зображень у відеоряді в режимі реального часу та їх розмірність, проведення таких зіставлень пов'язані з істотним обсягом обчислень. Крім того, значні області зображень часто виявляються малоінформативними або залежать від умов спостереження. Тому, для вирішення цих складностей науковцями запропоновано виділяти на еталонних зображеннях опорні ділянки, які є характерними об'єктами або регіонами з мінімальними визначеними змінами. Такий підхід дозволяє як скоротити обчислювальні витрати, так і підвищити надійність зіставлення за рахунок використання найбільш інформативних та стійких до змін фрагментів зображень.

Традиційні методи виявлення пропускають кожне зображення через попередньо встановлене вікно для вилучення ознак, а потім використовують навчений класифікатор для класифікації. Такі методи, як правило, вимагають багато людських ресурсів і зусиль для оброблення даних, і важко встановити єдині стандарти для ознак. Крім того, традиційні методи виявлення часто стикаються з такими проблемами, як висока часова складність, низька надійність і сильна залежність від сцени, що ускладнює їх практичне використання.

В останні роки, завдяки методам виявлення цілей, що постійно пропонуються на основі згорткових нейронних мереж (CNN), досягнуто певних позитивних результатів вирішення задачі виявлення. Залежність від обраних алгоритмів оброблення вхідного зображення формує двоетапні та одноетапні типи методів виявлення об'єктів, що мають переваги відносно високої точності виявлення та швидкості обчислень. Наприклад, згорткові мережі R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, Cascade R-CNN, R-FCN є двоетапними методами виявлення. Архітектурне рішення DenseBox, RetinaNet, серія SSD, серія YOLO є одноетапними.

Таким чином, даний розділ включає послідовний опис обраних методів (YOLO та Faster R-CNN) виявлення, розпізнавання та відстеження багатьох об'єктів за відеоданими з БпЛА в режимі реального часу та практичне порівняння обраних підходів та програмних рішень, результатом яких є розроблення технології для досягнення мети дисертаційного дослідження.

Оскільки швидкість виходу нових версій алгоритму YOLO дуже висока, на момент проведення порівняльної оцінки використовувались версії алгоритму v3 та v5, що на той момент були найновішими. Версію v3 можна вважати “офіційною” версією Ultralytics, головного вендора бібліотек цього алгоритму, а v5 з'явилась завдяки підтримці громади. При наявності на даний час досконаліших версій алгоритму YOLO, практичне порівняння обраних версій у даному розділі зумовлено конкретизацією параметрів якості та переваг по швидкості алгоритмів сімейства YOLO, які використані при виконанні практичної частини. На даний момент “офіційною” версією є v8.1 та вже розроблена версія v9, яка ще не є стабільною. Саме на версії v9 та на оптимізації та вдосконаленні її архітектурного рішення базується практична частина. Опис методу розпізнавання YOLO v9 включатиме опис алгоритму YOLO, склад функції втрат та топологію мережі в загальному виді.

Для покращення результатів розпізнавання через підвищення якості зображення розглянуто різні методи препроцесингу з їх математичним описом та алгоритмами.

Для подальшого використання в розробленій технології методу відстеження багатьох об'єктів BYTE Track у даному розділі приділена увага суттєвим для нього складовим: модель фільтрів Калмана та Угорський алгоритм для асоціації об'єктів.

2.1 Виявлення об'єктів у наборі даних Kitti при застосуванні YOLO та Faster R-CNN

Виявлення об'єктів має вирішальне значення для розвитку технологій автономного польоту, де складність і непередбачуваність реальних умов вимагають надійних і точних моделей.

В даному розділі розглянуто використання моделі Faster R-CNN для виявлення об'єктів з використанням набору даних Kitti з особливим акцентом на виявленні таких об'єктів, як дома, стадіони та перехрестя доріг. Завдяки всебічному навчанню і коригуванню моделі, Faster R-CNN було тонко налаштовано для адаптації до різноманітних викликів, які присутні у наборах даних.

У порівняльній оцінці використані YOLOv3 та YOLOv5, які слугували еталонами для визначення відносних переваг і недоліків Faster R-CNN. Результати експериментальних досліджень показали, що при досягненні високої точності алгоритм Faster R-CNN відставав у швидкості, що зробило моделі YOLO більш придатними для сценаріїв реального часу. Проведення порівняльного аналізу має на меті допомогти з вибором методів та алгоритмів побудови архітектури практичного рішення для вирішення задачі оптимізації моделей виявлення об'єктів для реальних застосувань на борту БпЛА.

2.1.1 Опис початкових даних

Результати апробації показують, що хоча Faster R-CNN і пропонує високий рівень точності, обчислювальні вимоги роблять модель менш придатною для сценаріїв, що вимагають негайної реакції. З іншого боку, моделі YOLO, які відомі своєю швидкістю обробки, пропонують більш оптимальне рішення для виявлення об'єктів у реальному часі, незважаючи на дещо нижчу точність порівняно з Faster R-CNN.

Останнім часом технології автономного польоту охоплюють ці моделі по-різному. На ринку автономного польоту продовжують випускати набори даних, які включають в себе як зображення, так і дані з датчиків, таких як LiDAR. Хоча багато з цих наборів даних надають анотації для об'єктів за допомогою обмежувальних рамок, лише декотрі з них розширюються до детального семантичного рівня, необхідного для більш детальних завдань, таких як те, що вирішується в даному дослідженні.

2.1.2 Опис набору даних Kitti

У дисертаційній роботі використовується набір даних Kitti з багатим набором складних реальних сцен, які точно імітують складнощі, що зустрічаються в різних умовах водіння чи польоту. Робота з таким реалістичним набором даних, дозволяє розширити межі стійкості та точності моделі в задачах виявлення об'єктів.

Колекція Kitti охоплює багатий набір типів даних, організованих в ієрархічній структурі, що сприяє вирішенню багатьох завдань комп'ютерного зору, зокрема, виявлення об'єктів (рисунок 2.1). Набір даних характеризується колекцією зображень високої роздільної здатності, які отримані у різних умовах польоту: від густонаселених міських районів до менш структурованих сільських доріг. Алгоритм виявлення систематично класифікує різні типи

об'єктів, кожен з яких ретельно анотований з точними 2D і 3D рамками. Анотації об'єктів пропонує розташування об'єктів у площині зображення та детальну інформацію про їх просторову орієнтацію та розміри в реальному контексті.

Набір даних Kitti має унікальну структуру, що складається з організованої колекції в zip-архівах, які розділені за датами і конкретними послідовностями руху. Сховище даних містить низку підкаталогів, пов'язаних з відео з камер, інформацією з GPS і датчиків руху, а також хмарами точок LiDAR. Все це має вирішальне значення для таких завдань, як сприйняття глибини і виявлення тривимірних об'єктів. Включення різнопланових сенсорних модальностей робить набір даних узагальненим.

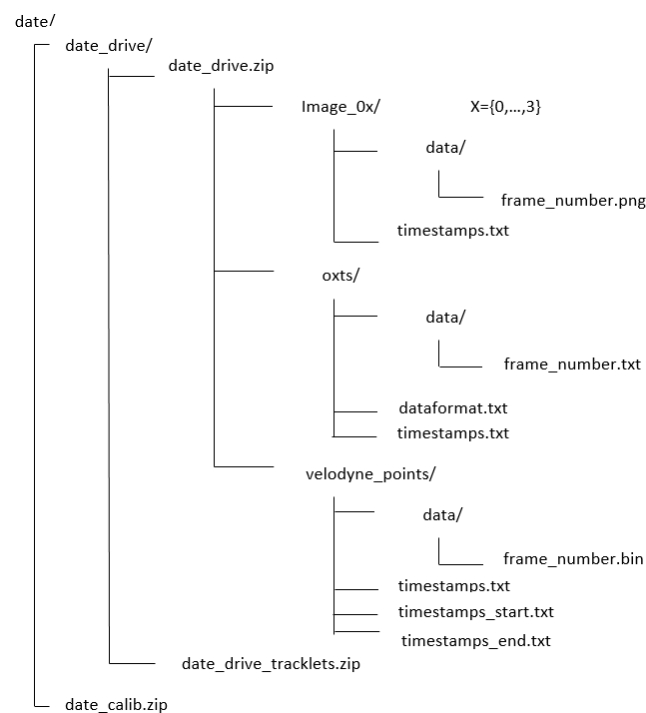


Рисунок 2.1 – Структура dataset Kitti

У архівах колекції Kitti набір даних є ретельно організованим і включає синхронізовані відеокадри, калібрувальні параметри для вирівнювання датчиків, треклети та пов'язані з ними анотації. Така детальна організація, включаючи часові мітки, калібрувальні дані для злиття датчиків і точне

маркування, необхідна для розробки алгоритмів, які покладаються на точну просторову інформацію, отриману з двовимірних зображень, для розуміння та інтерпретації тривимірного простору.

Включення в Kitti реальних сценаріїв польоту, різних умов освітлення, частих перешкод, значних змін ракурсу і безліч масштабів сприяє його репутації як суворого і надійного еталону для алгоритмів виявлення об'єктів. Архітектура набору даних і широке маркування спеціально розроблені для підвищення надійності алгоритмів комп'ютерного зору при забезпеченні їх ефективності у реальних умовах автономного польоту.

Таким чином, з моменту свого створення набір даних Kitti є важливим для еволюції технологій виявлення об'єктів та автономного управління дроном. Його реальні, різноманітні та складні сценарії гарантують, що будь-який алгоритм, перевірений на ньому, може відповідати різноманітним і суворим вимогам реальних ситуацій польоту, що робить його безцінним ресурсом як для дослідників, так і для практиків.

2.1.3 Порівняльна оцінка алгоритмів сімейства YOLO та Faster R-CNN

У величезному просторі виявлення об'єктів, що постійно розвивається, вибір метрик оцінювання та архітектури моделі відіграє ключову роль у визначенні ефективності підходу. У цьому розділі розглядаються методології, застосовані в поточному дослідженні, та висвітлюються нюанси оціночних та експериментальних процедур. Як основну метрику було використано усереднену точність mAP, що дає змогу комплексно оцінити ефективність виявлення на різних об'єктах за розміром та формою.

Для визначення можливостей та потенційних обмежень сучасних архітектур виявлення при аналізі звернено увагу на три моделі: Faster R-CNN, відому своїм ретельним механізмом пропозиції регіонів, та варіанти YOLO v3

і v5, відомі своєю швидкістю та можливістю застосування в реальному часі. Ці моделі ретельно протестовані на наборі даних виявлення об'єктів Kitti, що слугує еталонним тестом, який створює різноманітні проблеми та ситуації, типові для реальних сценаріїв польоту. За допомогою цієї методології надано уявлення про сильні сторони та потенційні напрямки вдосконалення кожної моделі в контексті виявлення об'єктів.

Результати роботи алгоритмів при використанні даних з колекції Kitti в моделях YOLOv3, YOLOv5 та Faster R-CNN наведено в таблицях 2.1-2.3 (визначення груп орієнтирів за розмірами наведено в розділі 2.1.7).

Таблиця 2.1 – Результати для Kitti з використанням модифікованого YOLO v3.

Орієнтир	Легкий	Середній	Складний
Автомобіль	56,0%	36,23%	29,55%
Пішохід	29,98%	22,84%	22,21%
Велосипедист	9,09%	9,09%	9,09%

Таблиця 2.2 – Результати для Kitti з використанням модифікованого YOLO v5.

Орієнтир	Легкий	Середній	Складний
Автомобіль	88,17%	78,70%	69,45%
Пішохід	60,44%	43,69%	43,06%
Велосипедист	55,00%	39,29%	32,58%

Таблиця 2.3 – Результати для Kitti з використанням Faster R-CNN

Орієнтир	Легкий	Середній	Складний
Автомобіль	84,81%	86,18%	78,03%
Пішохід	76,52%	59,98%	51,84%
Велосипедист	74,72%	56,83%	49,60%

2.1.4 Ключові показники для оцінки

Для всебічної оцінки ефективності моделей виявлення об'єктів дуже важливо розглянути показники, які охоплюють як точність, так і повноту.

Усереднена точність (mAP) є комплексною метрикою, яка відображає точність алгоритму для різних класів і визначається в термінах точності і функції пам'яті.

Точність (Precision, P) є відношенням кількості правильно передбачених позитивних спостережень до загальної кількості передбачених позитивних спостережень:

$$P = \frac{TP}{TP+FP} \quad (2.1)$$

де

TP – кількість правильно передбачених позитивних спостережень (істинно позитивні),

FP – загальна кількість передбачених позитивних спостережень (хибно позитивні результати).

Повнота (Recall, R) є відношенням правильно передбачених позитивних спостережень до всіх спостережень у фактичному класі:

$$R = \frac{TP}{TP+FN} \quad (2.2)$$

де FN – хибно негативні результати.

Для фіксованого класу абсолютна точність AP (Absolute Precision) є площею під кривою точність-відгук $P(R)$, яка відображає точність як функцію відгуку. Розрахунок середньої точності наведено у формулі:

$$AP = \int_0^1 P(R) dR \quad (2.3)$$

На практиці крива $P(R)$ є дискретною, і точність AP розраховується за допомогою чисельних методів, як середньозважене значення точності на кожному пороговому значенні, причому в якості ваги використовується

приріст пригадування від попереднього порогового значення. Середньозважене значення точності розраховується за формулою:

$$AP = \sum_{n=1}^N (R_n - R_{n-1})P_n \quad (2.4)$$

де

P_n – точність на n -му порозі,

R_n – повнота на n -му порозі.

При використанні такого підходу похибка mAP тоді є середнім значенням середньої точності по всіх класах та її розрахунок наведено у формулі:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.5)$$

де

N – це кількість класів,

AP_i – середня точність для i -го класу.

Показник $mAP_{0.5}$ позначає середню точність, коли IoU моделі виявлення дорівнює 0.5, а $mAP_{0.5:0.95}$ позначає середню точність, коли IoU моделі виявлення дорівнює від 0.5 до 0.95 (зі значеннями, взятими з інтервалом 0.5). Така метрика ефективно відображає загальну продуктивність алгоритму виявлення об'єктів у всіх класах, пропонуючи єдину узгоджену метрику, яка враховує існування і точну локалізацію об'єктів.

На практиці показник mAP обчислюється спочатку шляхом впорядкування виявлених об'єктів у всіх класах за їх показниками достовірності. В алгоритмі надалі враховуються істинно позитивні (True Positive, TP) і хибно позитивні (False Positive, FP) виявлення на різних рівнях точності, які, як правило, збільшуються невеликими кроками, поки не буде досягнуто значення точності, що дорівнює 1.

На кожному рівні обчислюється точність P та показник AP . Метрика mAP є особливо ваговою в сценаріях з незбалансованими наборами даних, де декотрі класи є недостатньо представленими. При отриманні середнього

значення між класами показник mAP гарантує, що всі класи роблять будуть робити однаковий внесок у загальну ефективність моделі незалежно від їх частоти.

2.1.5 Деталі навчання для Faster R-CNN

Щоб використати весь потенціал Faster R-CNN для набору даних Kitti, було застосовано багатоетапний підхід до навчання. Кожен етап був ретельно розроблений для поступового покращення продуктивності моделі на складних реальних даних, представлених колекцією Kitti.

Алгоритм навчання Faster R-CNN наведено на рисунку 2.2.

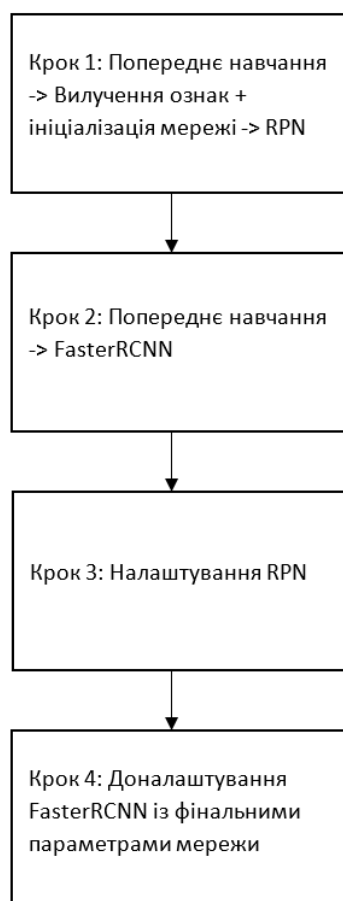


Рисунок 2.2 – Алгоритм навчання Faster R-CNN

Етап 1: Ініціалізація мережі регіональних пропозицій (RPN). Ініціалізовано RPN при використанні ваги з моделі, яка попередньо навчена на комплексному наборі даних включно з перенесенням досвіду, щоб скористатися попередньо вивченими особливостями. Ця стратегія прискорила процес збіжності та покращила здатність RPN генерувати високоякісні пропозиції, пристосовані до унікальних характеристик набору даних Kitti.

Етап 2: Навчання Faster R-CNN. На цьому етапі було спрямовано зусилля на навчання детектора Faster R-CNN. При використанні пропозицій, які згенеровані RPN, було навчено детектор класифікувати об'єкти у запропонованих областях та проводити уточнення їх обмежувальних рамок. На цьому етапі було використано нейронну згорткову мережу, що дозволило адаптувати ваги моделі до специфічних нюансів набору даних Kitti і при цьому підвищити точність розпізнавання.

Етап 3: Налаштування RPN. Спираючись на експериментальні результати роботи Faster R-CNN, було проведено повернення до етапу 1 (RPN для подальшого донавчання. Мета введення цього етапу полягала в тому, щоб ітераційно покращити здатність мережі пропонувати регіони, спираючись на спільні шари згортки, які є оптимізованими для виявлення об'єктів що специфічні для Kitti. Зберігаючи ці шари фіксованими, забезпечено стабільність вилучення ознак, при одночасному відточуванні механізму пропозицій RPN для підвищення точності.

Етап 4: Фінальна настройка детектора. На останньому етапі було доналаштовано детектор Faster R-CNN з подвійним фокусом на використанні знайдених пропозицій від RPN та збереженні цілісності спільних згорткових ознак. Цей етап був критично важливим для досягнення тонкого балансу між якістю пропозицій та здатністю виявлення, що є ключовим для високої продуктивності на наборі даних Kitti.

Робочий процес включав налаштування TensorFlow Object Detection API, перетворення набору даних Kitti у формат TensorFlow tfrecord, а також

ретельний процес навчання та тестування. На кожній ітерації ретельно проведено відстеження метрики втрат і візуалізації прогнозів граничних областей, щоб ітеративно підлаштовувати модель під поріг втрат нижче 0.1, що свідчить про добре навчену модель Faster R-CNN для задачі виявлення об'єктів Kitti.

2.1.6 Деталізація етапів навчання для YOLOv3 та YOLOv5

Для використання потенціалу моделі YOLO використані наступні етапи [23].

Етап 1: Адаптація до набору даних Kitti. Стратегія навчання для варіантів алгоритму YOLO була точно налаштована на унікальну структуру набору даних Kitti. YOLOv3 та YOLOv5 було адаптовано для роботи з широким співвідношенням сторін зображень. Було уникнуто спотворення обмежувальних рамок, змінивши вхідні параметри роздільної здатності відповідно до власного співвідношення сторін набору даних, що забезпечило цілісність процесу розпізнавання об'єктів.

Етап 2: Оптимізація конфігурації. Обидві моделі YOLO пройшли серію конфігураційних коригувань для оптимізації архітектури мережі для набору даних Kitti. Ці оптимізації включали специфічні зміни розміру вхідних даних для збереження початкового співвідношення сторін зображень та коригування розмірів фільтрів у шарах розпізнавання для точного відображення кількості класів. В моделі на цьому етапі вимкнено випадкову зміну розмірів для забезпечення стабільних та узгоджених вхідних даних для моделей.

Етап 3: Налаштування гіперпараметрів. Подальше налаштування гіперпараметрів було проведено для покращення продуктивності моделей. Швидкість навчання, масштаби об'єктів та пороги були відкалібровані для оптимальної точності розпізнавання. Цей процес точного налаштування був критично важливим для забезпечення того, щоб моделі були пристосовані не

лише до розмірів зображень набору даних, але й до характеристик об'єктів. Ці коригування відіграли важливу роль у визначенні продуктивності YOLOv3 та YOLOv5 [32] на наборі даних Kitti, що призвело до покращення моделі виявлення об'єктів, яка є одночасно точною та ефективною.

2.1.7 Результати досліджень та порівняльний аналіз

На рисунку 2.3 представлено показові результати тестування, отримані за допомогою трьох моделей, що розглядаються: Faster R-CNN, YOLOv3 та YOLOv5. Для цієї емпіричної оцінки обрано три дорожні сценарії з набору даних Kitti, які характеризуються великою кількістю автомобілів, пішоходів та різноманітним набором класів об'єктів. Моделі було протестовано на трьох різних наборах даних: легкий, середній та складний. В легкому наборі даних шуканий об'єкт видно повністю, в складному – видно лише його частину. Середній набір містить кількість шуканих об'єктів з легкого та складного наборів, що формується у співвідношенні 1:1 відповідно.

Як показано на рисунку 2.3, модель Faster R-CNN демонструє найкращі результати порівняно з обома варіантами YOLO. У визначеному експерименті моделі YOLO демонструють помітні обмеження у виявленні людей, розташованих з лівого боку, і здатні ідентифікувати лише одного пішохода праворуч. На відміну від них, модель Faster R-CNN вміло розпізнає декілька пішоходів праворуч, що підкреслює його покращену здатність виявлення об'єктів в складних умовах.

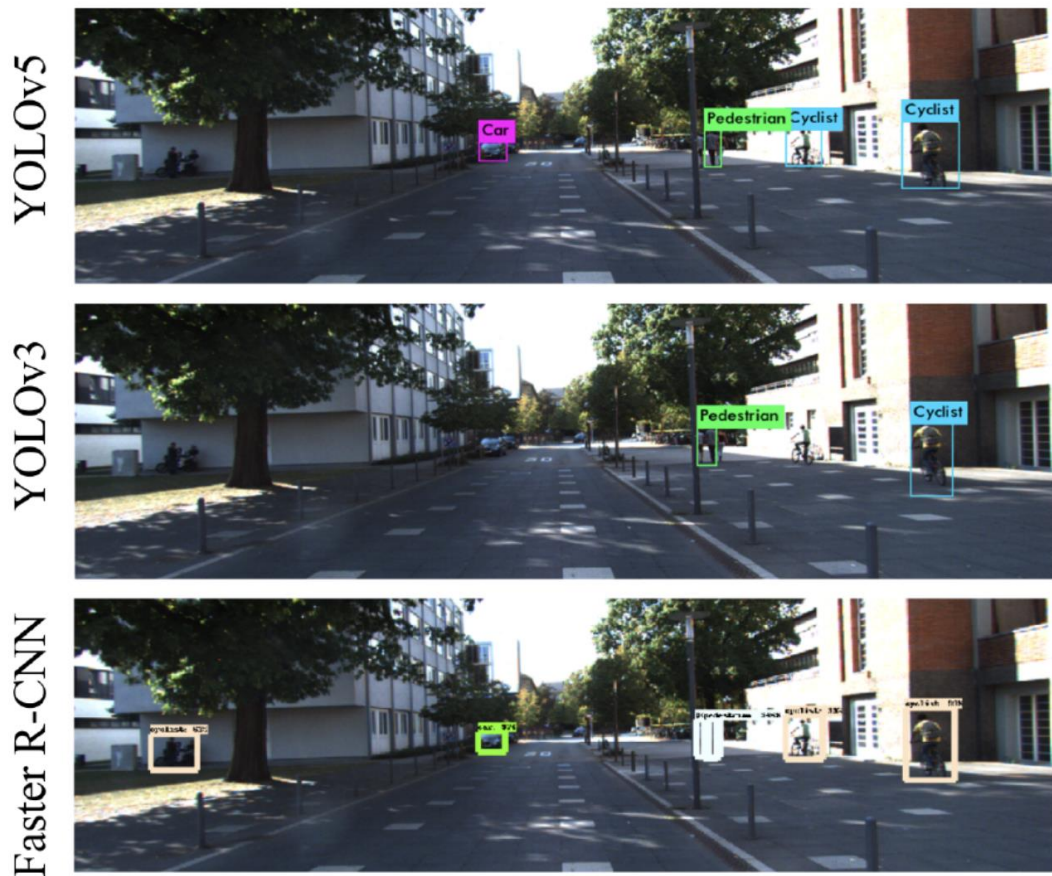


Рисунок 2.3 – Результати тестування, отримані за допомогою Faster R-CNN, YOLOv3 та YOLOv5

Комплексна кількісна оцінка при використанні підходу поділу на легкі, помірні та складні набори даних виявила глибокі закономірності в роботі алгоритмів розпізнавання. У найпростішому наборі даних, де умови були найбільш сприятливими (рисунок 2.4), Faster R-CNN продемонстрував виняткову точність виявлення дорожніх розв'язок, досягнувши 88,17% порівняно з 56,00% для YOLOv3 і 84,81% для YOLOv5. Однак для пішоходів YOLOv5 перевершив свої аналоги з показником 60%, тоді як Faster R-CNN показав лише 29,98%, що підкреслює мінливість результатів для різних категорій об'єктів.

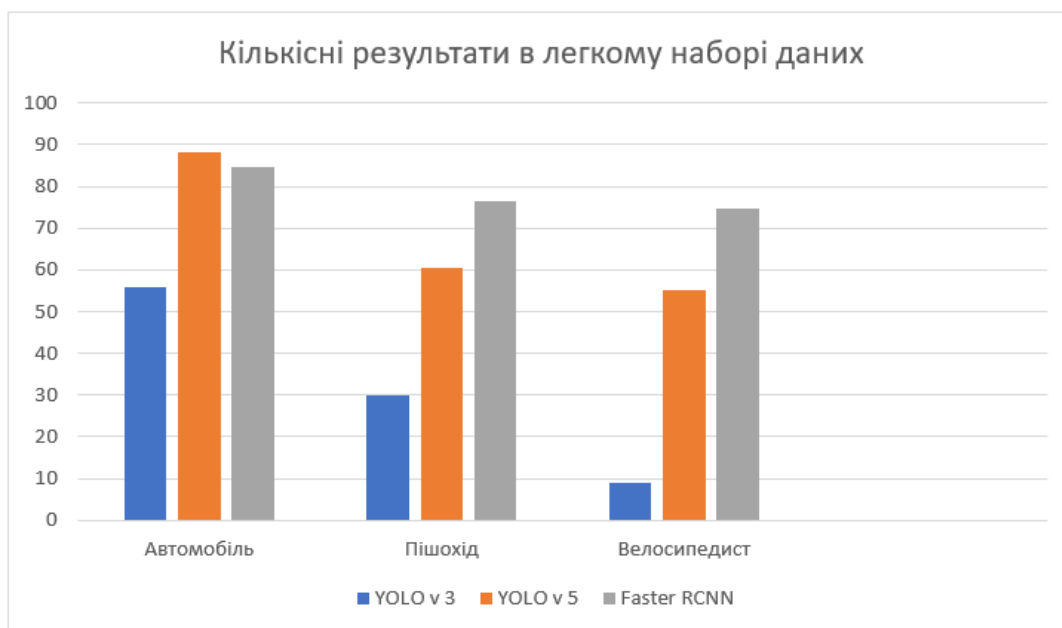


Рисунок 2.4 – Тестові приклади методів YOLOv3, YOLOv5, FasterRCNN в легкому наборі даних

Коли складність набору є середньою (рисунок 2.5), Faster R-CNN зберіг лідерство у виявленні дорожніх розв'язок на рівні 86,18%, що в порівнянні з простим набором є лише незначним зниженням, демонструючи його стійкість до мінливих умов. На противагу цьому, YOLOv3 і YOLOv5 зазнали більш значного падіння до 36,23% і 78,70% відповідно. Ця тенденція збереглася і при виявленні велосипедистів, де Faster R-CNN показав найкращий результат – 56,83%, тоді як YOLOv5 утримався на рівні 59,98%, а YOLOv3 значно почав відставати від нього, отримавши лише 9,09% виявлення.

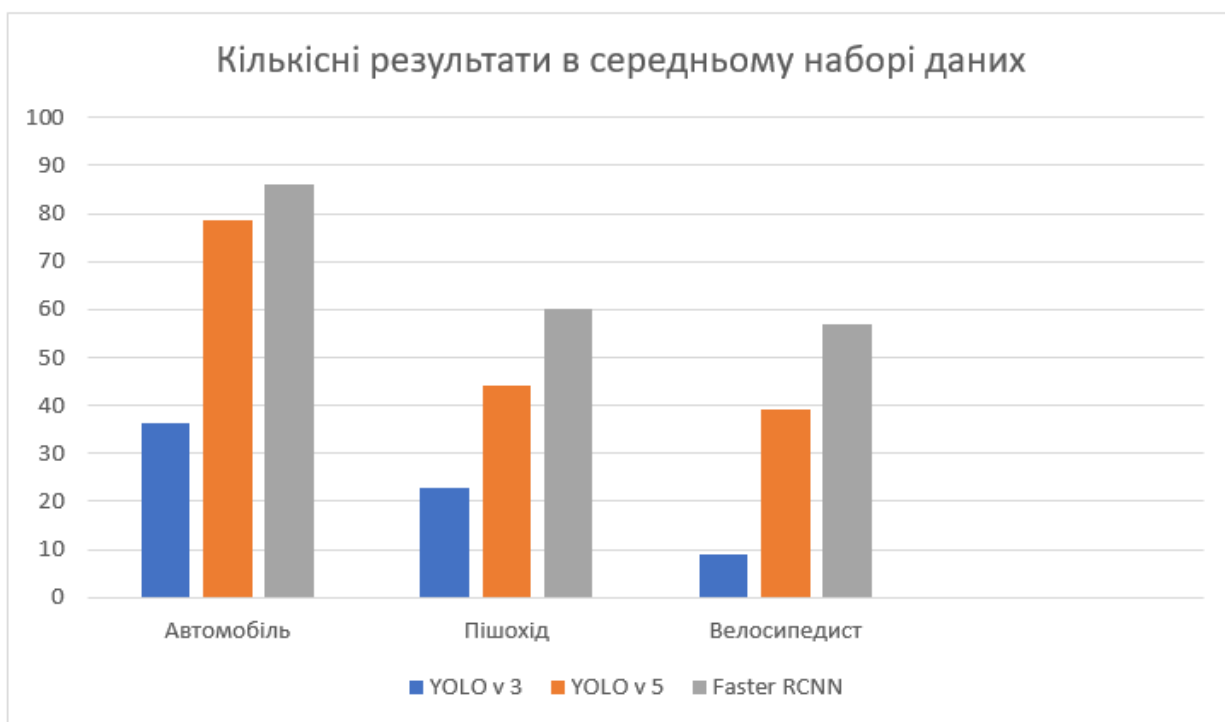


Рисунок 2.5 – Тестові приклади методів YOLOv3, YOLOv5, Faster RCNN в середньому наборі даних

Складний набір даних (рисунок 2.6), що характеризує найскладніші сценарії, продовжує підкреслювати гарну продуктивність Faster R-CNN, особливо при виявленні автівок з успішністю 78,03%. Показники YOLOv5 також заслуговують на увагу, особливо при виявленні авто – 43%, що не далеко відстає від показника Faster R-CNN (51.84%). Однак ефективність YOLOv3 різко впала в усіх категоріях, досягнувши 29,55% для авто і 9,09% для пішоходів і велосипедистів.

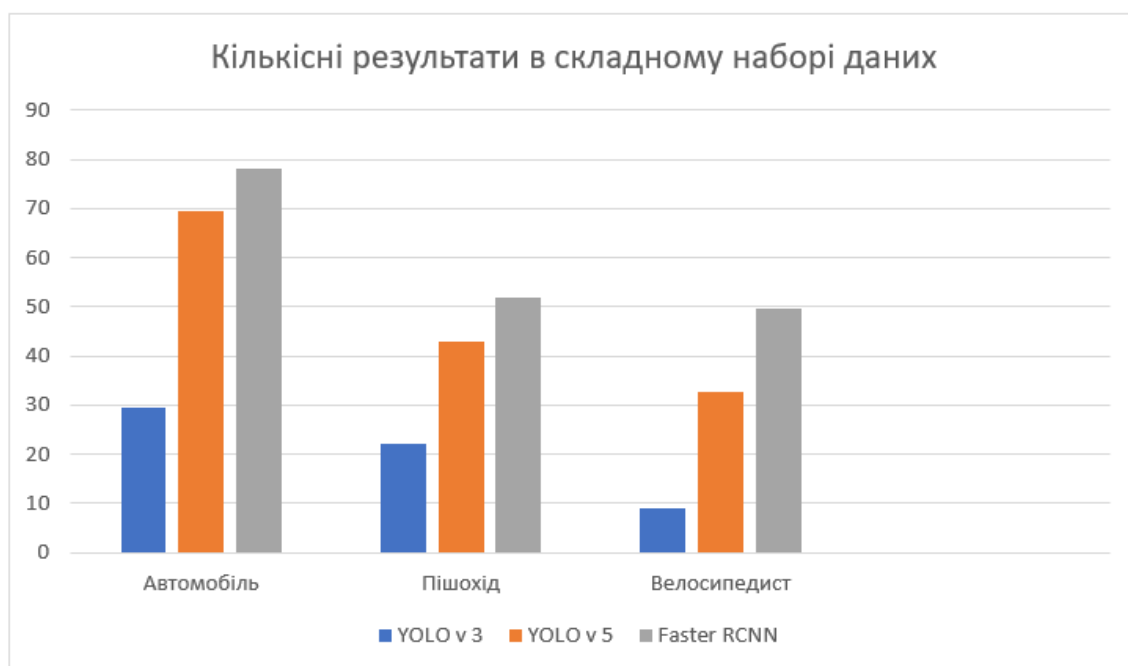


Рисунок 2.6 – Тестові приклади методів YOLOv3, YOLOv5, FasterRCNN в складному наборі даних

Стабільність результатів Faster R-CNN на всіх рівнях складності підкреслює його надійні можливості виділення ознак та пропозиції регіонів, особливо при обробленні складних сцен з різноманітними об'єктами. І навпаки, хоча YOLOv5 продемонстрував значну адаптивність, перевершивши YOLOv3 і навіть Faster R-CNN у деяких випадках, його результати не були такими ж послідовними для різних категорій і наборів даних. Ці спостереження вказують на компроміс між високошвидкісною обробкою архітектури YOLO і ретельною обробкою на основі регіонів Faster R-CNN, яка, хоча і працює повільніше, але забезпечує підвищену точність у завданнях виявлення об'єктів.

Отже, аналіз на основі даних підтверджує, що Faster R-CNN є більш надійною і точною моделлю для виявлення об'єктів у різноманітних і складних середовищах, що є критичною вимогою для систем автономного польоту. Стабільна продуктивність в різних умовах і категоріях об'єктів робить її

кращою моделлю для застосувань, де надійність і точність мають першочергове значення.

2.1.8 Аналіз часу виконання алгоритмів моделей

В доповнення до оцінки точності виявлення, було проведено оцінку часу виконання алгоритмів кожної з трьох моделей при опрацюванні даних зображення.

Таблиця 2.4 – Час оброблення зображення

Модель	YOLOv3	YOLOv5	Faster R-CNN
Час	15 мс	35 мс	2763 мс

Як YOLOv3, так і YOLOv5 позиціонуються як фреймворки для виявлення в реальному часі. Експерименти підтвердили це твердження, оскільки для набору даних Kitti обидві моделі змогли завершити виявлення об'єктів менш ніж за 40 мс на зображення. Причому YOLOv5 продемонстрував незначне відставання в часі порівняно зі своїм попередником, YOLOv3.

На противагу цьому, модель Faster R-CNN, незважаючи на свою заявлену швидкість, продемонструвала значно більший час виконання порівняно з серією YOLO. Такий час роботи робить Faster R-CNN менш придатною для застосувань, що вимагають реакції в реальному часі, таких як автономний політ. Хоча Faster R-CNN випереджає моделі YOLO за точністю виявлення, її швидкість ставить під сумнів доцільність її використання для критично важливих завдань, пов'язаних з часом. Це підкреслює важливість ретельного вибору відповідних моделей на основі специфічних вимог і обмежень конкретного застосування.

2.2 Моделі та реалізації алгоритму YOLO

Враховуючи описані вище результати порівняльного аналізу, для подальшої роботи над побудовою програмного рішення було обрано групу алгоритмів YOLO, оскільки попри менш високу точність розпізнавання його швидкодія на практиці набагато перевищує показники Faster R-CNN. За час написання дисертаційної роботи вийшло декілька нових версій алгоритму YOLO, таких як YOLOv7, YOLOv8 та YOLOv9, які попри мінімальні втрати у швидкості значно покращують показники розпізнавання на різноманітних наборах даних. На жаль, за цей час не було анонсовано нових версій алгоритмів сімейства R-CNN, тому можна вважати результати попереднього пункту релевантними, та такими, що відображають поточні тренди в сфері розпізнавання об'єктів.

Особливою характеристикою технології YOLO є надзвичайна швидкість її роботи, що дозволило використовувати її в задачах розпізнавання об'єктів у реальному часі. Ще однією перевагою даного підходу є гнучкість до опрацювання зображень різної розмірності та розпізнавання різної кількості класів без значних змін в архітектурі моделі. Також, в моделях YOLO було частково вирішено проблему розпізнавання об'єктів при різних масштабах.

Переваги YOLO та великий модифікаційний потенціал стали причиною стрімкого розвитку даного підходу для вирішення багатьох задач: від сегментації об'єктів й до відстеження багатьох об'єктів у реальному часі. Тому, наразі більш коректне визначення для YOLO, як сімейство моделей об'єднаних єдиними алгоритмічними підходами загальної концепції розпізнавання за один прохід.

2.2.1 Алгоритмічна основа

Загальний алгоритм роботи YOLO працює на основі наступних чотирьох базових підходів:

- поділ на зображення на блоки (Residual blocks);
- обмежувальні рамки (Bounding box regression);
- перетин над об'єднанням (Intersection Over Unions або скорочено IOU);
- немаксимальне стримування (Non-Maximum Suppression).

Поділ на зображення на блоки. Перший крок починається з поділу вихідного зображення I на $N \times N$ клітинок сітки однакової форми, де N у нашому випадку дорівнює 8 (як на першому масштабі моделі YOLOv2) і як показано на рисунку 2.7. Кожна клітинка в сітці на подальших кроках відповідає за локалізацію та прогнозування класу об'єкта, який вона охоплює, а також значення ймовірності/достовірності.



Рисунок 2.7 – Розбиття зображення на $N \times N$ блоків

Наступним кроком є визначення обмежувальних рамок (рисунок 2.8), які відповідають прямокутникам, що виділяють всі об'єкти на зображенні [33].

Можна мати стільки обмежувальних рамок, скільки є об'єктів на зображенні, але у моделі є алгоритмічне обмеження на кількість рамок для 1 клітинки поділу (залежить від версії, наприклад в YOLOv1 їх може бути 2 для кожного з 49 блоків-клітинок, а у YOLOv8 потенційно їх тисячі загалом, в залежності від заданого розміру моделі).

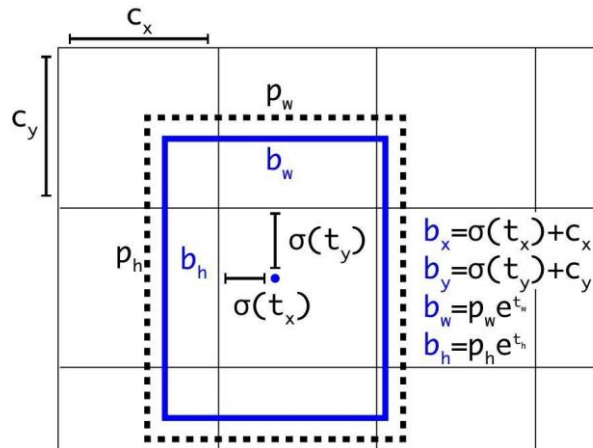


Рисунок 2.8 – Обмежувальна рамка з попередніми розмірами і прогнозуванням розташування

Модель YOLO визначає атрибути цих обмежувальних рамок за допомогою одного модуля регресії у наступному форматі:

$$Y = [B_x, B_y, B_h, B_w, P_0, C_1, C_2, C_3, \dots, C_n] \quad (2.6)$$

де

Y – це кінцеве векторне представлення для кожного обмежувальної рамки;

P_0 – відповідає оцінці ймовірності блоку сітки, що містить об'єкт; наприклад, всі сітки, які позначені червоним кольором, матимуть значення ймовірності більше нуля; приклад на рисунку 2.9 є спрощеною версією, оскільки ймовірність кожної зеленої клітинки близька до 0 (є незначною);

B_x, B_y – координати x та y центру обмежувальної рамки відносно клітинки сітки, що її огинає.

B_h, B_w – відповідають висоті і ширині обмежувальної рамки по відношенню до блоку-клітинки сітки, що її охоплює.

$C_1, C_2, C_3, \dots, C_n$ – відповідають класам, які модель передбачає.

Кількість класів не обмежується базовою архітектурою YOLO, а визначається конкретною задачею.



Рисунок 2.9 – Приклад зображення з виділеним об'єктом та блоками сітки, де оцінка ймовірності наявності об'єкта більше відповідного параметру

Перетин над об'єднанням (Intersection Over Unions). Більше одного об'єкту на зображенні може мати кілька ймовірних обмежувальних рамок на прогнозування, навіть якщо не всі з них є релевантними. Метрика Intersection Over Unions (IOU) визначає міру схожості двох обмежувальних рамок між собою через відношення площі перетину до площі об'єднання двох рамок (рисунок 2.10). Метою використання метрики IOU ($IOU \in [0, 1]$) є відкидання не релевантних клітинок сітки та обмежувальних рамок.

Логіка такого підходу реалізує стратегію того, що користувач визначає поріг відбору IOU (стандартно дорівнює 0.5) і потім алгоритм YOLO обчислює IOU кожної клітинки сітки (дорівнює відношенню площі перетину до площі об'єднання). Надалі алгоритм повинен проігнорувати прогноз комірок сітки,

що мають $IOU \leq$ порогового значення, і врахувати ті, що мають $IOU >$ порогового значення (threshold).

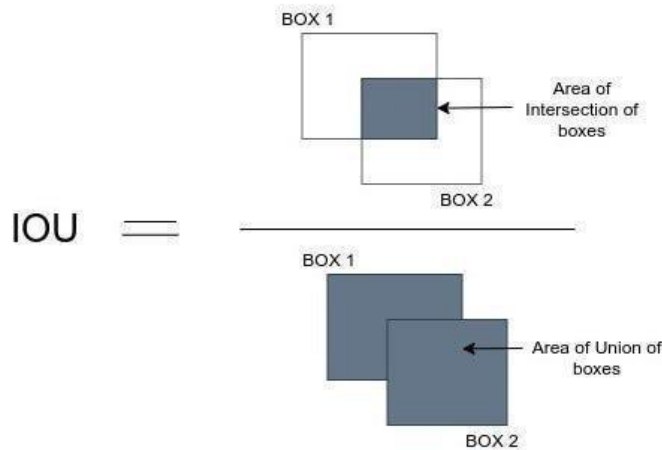


Рисунок 2.10 – Схема розрахунку IOU

Немаксимальне стримування (Non-Maximum Suppression). Встановлення порогу для IOU не завжди достатньо, оскільки об'єкт може мати кілька обмежувальних рамок з IOU, що перевищують мінімальний поріг. Залишення всіх цих обмежувальних рамок може спричинити шум і похибки (коли один об'єкт буде розпізнано як декілька). Для уникнення цього алгоритм YOLO використовує немаксимальне стримування, щоб залишити тільки ті обмежувальні рамки, які мають найвищу ймовірність виявлення об'єкту:

$$B_0 = P_0(B_i) \quad (2.7)$$

де

B_0 – фінальна обмежувальна рамка для об'єкта,

B_i – потенційні рамки.

2.2.2 Еволюція архітектури мережі YOLO

Найбільшою інноваційною на момент написання роботи архітектурою мережі є YOLOv9 (рисунок 2.11), яка вийшла в квітні 2024 року й стала

проривом у сфері розпізнавання об'єктів у реальному часі, анонсувавши нові алгоритми, які змінюють правила гри, а точніше програмовану інформацію про градієнт (Programmable Gradient Information, PGI), та узагальнену ефективну мережу агрегації рівнів (Generalized Efficient Layer Aggregation Network, GELAN). Дана модель алгоритму демонструє вагомі покращення в гнучкості, ефективності та точності на визначеному наборі даних MS COCO. Реалізувала її окрема команда розробників, яка розмістила відкритий код з модулями YOLOv9 у публічний GitHub репозиторій.

На момент описання архітектури YOLOv9 в цій роботі команда Ultralytics не опублікували жодної наукової роботи або статті з детальним описом нової архітектури, обґрунтуванням модифікацій та коментарями авторів. Тому розділ буде базуватися на загальному описі еволюції архітектури YOLO з моменту версії YOLOv5, оскільки саме вона використовувалась в попередньому розділі при порівнянні з Faster R-CNN.

У свою чергу, модель YOLOv8, реалізована безпосередньо командою Ultralytics стала вінцем еволюції (State of the art, SOTA) для задач розпізнавання об'єктів та розпізнавання об'єктів у реальному часі. Попри невеликий час існування модель встигла зарекомендувати себе золотим стандартом, яка досягає високих показників по ключовим параметрам.

2.2.2.1 Модель YOLO v8

Порівняно з попередніми моделями серії YOLO (такими як YOLO v5 та YOLO v7), алгоритм YOLO v8, який випущений компанією Ultralytics 10 січня 2023 року, є вдосконаленою та передовою моделлю, що пропонує вищу точність та швидкість виявлення. Структура мережі YOLO v8 представлена з хребта, шиї та голови, як показано на рисунку 2.11.

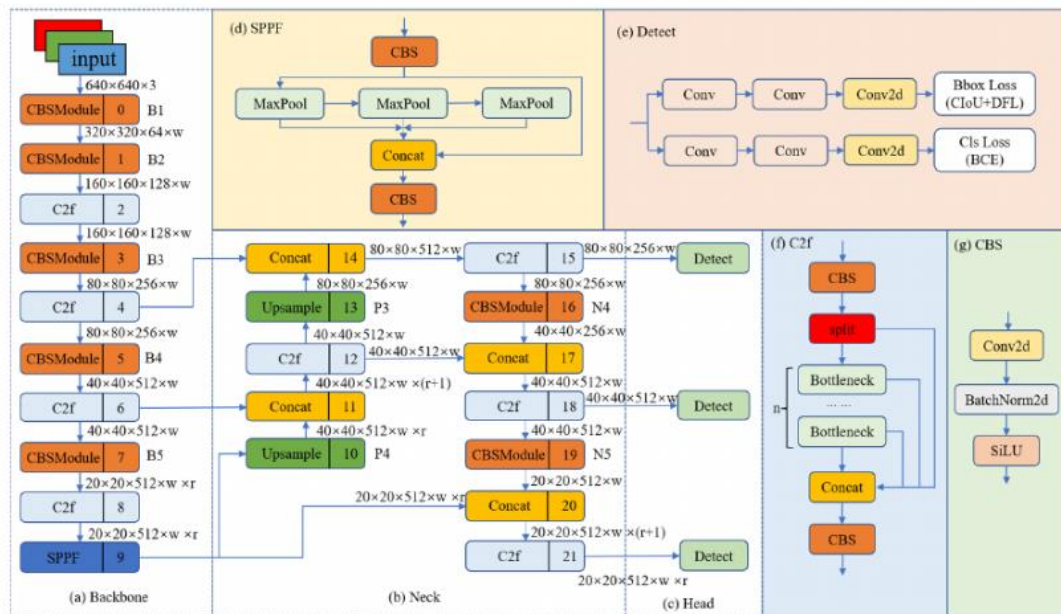


Рисунок 2.11 – Структура мережі YOLO v8

Хребет (backbone) YOLO v8 використовує модифіковану CSPDarknet53 [34] як опорну мережу, а вхідні ознаки дискретизуються п'ять разів для отримання п'яти ознак різного масштабу, які позначаємо як B1-B5. Структуру опорної мережі показано на рисунку 2.14 у зоні (a). Модуль Cross Stage Partial (CSP) в оригінальній опорній мережі замінено на модуль C2f, і структура модуля C2f показана на рисунку 2.14 у зоні (f), де n позначає кількість вузьких місць. Модуль C2f використовує градієнтне шунтове з'єднання для збагачення інформаційного потоку мережі вилучення ознак, зберігаючи при цьому невелику вагу. Модуль CBS виконує операцію згортки вхідної інформації з подальшою пакетною нормалізацією і, нарешті, активує інформаційний потік за допомогою SiLU для отримання вихідного результату, як показано на рисунку 2.14 у зоні (g).

Магістральна мережа використовує модуль швидкого просторового пірамідального об'єднання (spatial pyramid pooling fast, SPPF) для об'єднання вхідних карт об'єктів у карту фіксованого розміру для адаптивного виведення. Порівняно зі структурою просторового пірамідального об'єднання (spatial pyramid pooling, SPP) [35], SPPF зменшує обчислювальні зусилля і має меншу

затримку завдяки послідовному з'єднанню трьох максимальних шарів об'єднання, як показано на рисунку 2.14 у зоні (d).

При врахуванні наробок PANet [36], мережа YOLOv8 розроблена зі структурою PAN-FPN на ший (neck), як показано на рисунку 2.14 у зоні (b). При порівнянні зі структурою ший моделей YOLOv5 і YOLOv7, модель YOLOv8 усуває операцію згортки після підвищення дискретизації в структурі PAN, що зберігає початкову продуктивність при спрощенні моделі. P4-P5 та N4-N5 використовуються для позначення двох різних шкал ознак у PAN-структурі та FPN-структурі моделі YOLOv8 відповідно. Традиційно FPN використовує підхід "зверху вниз" для передачі глибокої семантичної інформації. FPN покращує семантичну інформацію ознак шляхом об'єднання B4-P4 та B3-P3, але при цьому втрачається частина інформації про локалізацію об'єкта. Щоб вирішити цю проблему, PAN-FPN додає PAN до FPN. PAN покращує вивчення інформації про місцезнаходження, об'єднуючи P4-N4 і P5-N5 для реалізації покращення шляху в низхідній формі. PAN-FPN буде мережеву структуру зверху вниз і знизу вгору, яка реалізує взаємодоповнюваність поверхневої позиційної інформації та глибокої семантичної інформації за допомогою злиття ознак, що призводить до різноманітності та повноти ознак.

Частина виявлення моделі YOLOv8 використовує відокремлену структуру голови, як показано на рисунку 2.14 у зоні (e). У відокремленій структурі голови використовуються дві окремі гілки для класифікації об'єктів і прогнозування регресії граничного поля. Для цих двох типів задач застосовуються різні функції втрат. Для задачі класифікації використовується бінарна перехресна ентропійна втрата (binary cross-entropy loss, BCE Loss). Для задачі регресії з обмеженням прогнозованого поля використовуються розподілені фокальні втрати (distribution focal loss, DFL) [37] та CIoU [38]. Така структура виявлення дозволяє підвищити точність виявлення і прискорити збіжність моделі. Модель YOLOv8 позиціонується як модель детектування без

якоря, яка стисло визначає позитивні та негативні зразки. Модель YOLOv8 також використовує Task-Aligned Assigner [39] для динамічного призначення зразків, що покращує точність виявлення та робастність моделі.

2.2.2.2 Модель YOLOv9

Даний розділ базується на загальному описі моделі YOLOv9 на сайті розробника та зворотної інженерії (reverse engineering) коду модулів з публічного GitHub репозиторію. Архітектура складається з двох основних частин: хребта (backbone) та голови (head), в попередніх версіях моделі була ще частина neck, що в YOLOv9 об'єднана з head.

Перш ніж розглядати мережеву структуру YOLOv9 необхідно взяти до уваги технологію перепараметризації (re-parameter), яка є відомою технологією (наприклад, в операції FuseCon алгоритму YOLOv5).

Перепараметризація є мережевою структурою, яка оптимізована з точки зору ефективності та продуктивності мережевих архітектур. Основна ідея перепараметризації полягає у використанні кількох гілок (наприклад кількох шарів Conv) під час навчання, щоб збільшити шлях градієнтного зворотнього зв'язку. Злиття виконується під час фази висування висновків, для зменшення кількості обчислень і підвищення ефективності розрахунків.

При розгляді згортки (Conv) та пакетної нормалізації (Batch Normalization, BatchNorm2d, BN) стає зрозумілим, що шар згортки насправді є операцією виду $y = ax + b$.

На етапі навчання основна операція складається з двох частин: згортка та пакетна нормалізація. Операція згортки наведена у формулі:

$$x = conv.weight * xconv.bias, \quad (2.8)$$

де

$conv.weight$ – вага згортки;

$xconv.bias$ – зміщення згортки.

Пакетна нормалізація наведена у формулі:

$$x = bn.\gamma * \frac{x_i - bn.mean}{\sqrt{bn.var + bn.\epsilon}} + bn.\beta \quad (2.9)$$

де

bn.mean – середнє значення (відповідає *running_mean* у *BatchNorm2d*);

bn.var – це дисперсія (відповідає *running_var* у *BatchNorm2d*);

bn.γ, *bn.β* – відповідають вазі та зсуву у *BatchNorm2d* відповідно;

bn.ε – епсілон (*eps* у *BatchNorm2d*).

На етапі формування висновків можна відповідно консолідувати Conv і BN, у той же час параметри потрібно повторно зіставити наступним чином:

$$x = \frac{bn.\gamma * conv.weight}{\sqrt{bn.var + bn.\epsilon}} * x + \frac{bn.\gamma * conv.bias}{\sqrt{bn.var + bn.\epsilon}} + bn.\beta \quad (2.10)$$

Як і вище введені коефіцієнти формують нову вагу та нове зміщення відповідно.

Загальна архітектура YOLO v9 проілюстрована на рисунку 2.12.

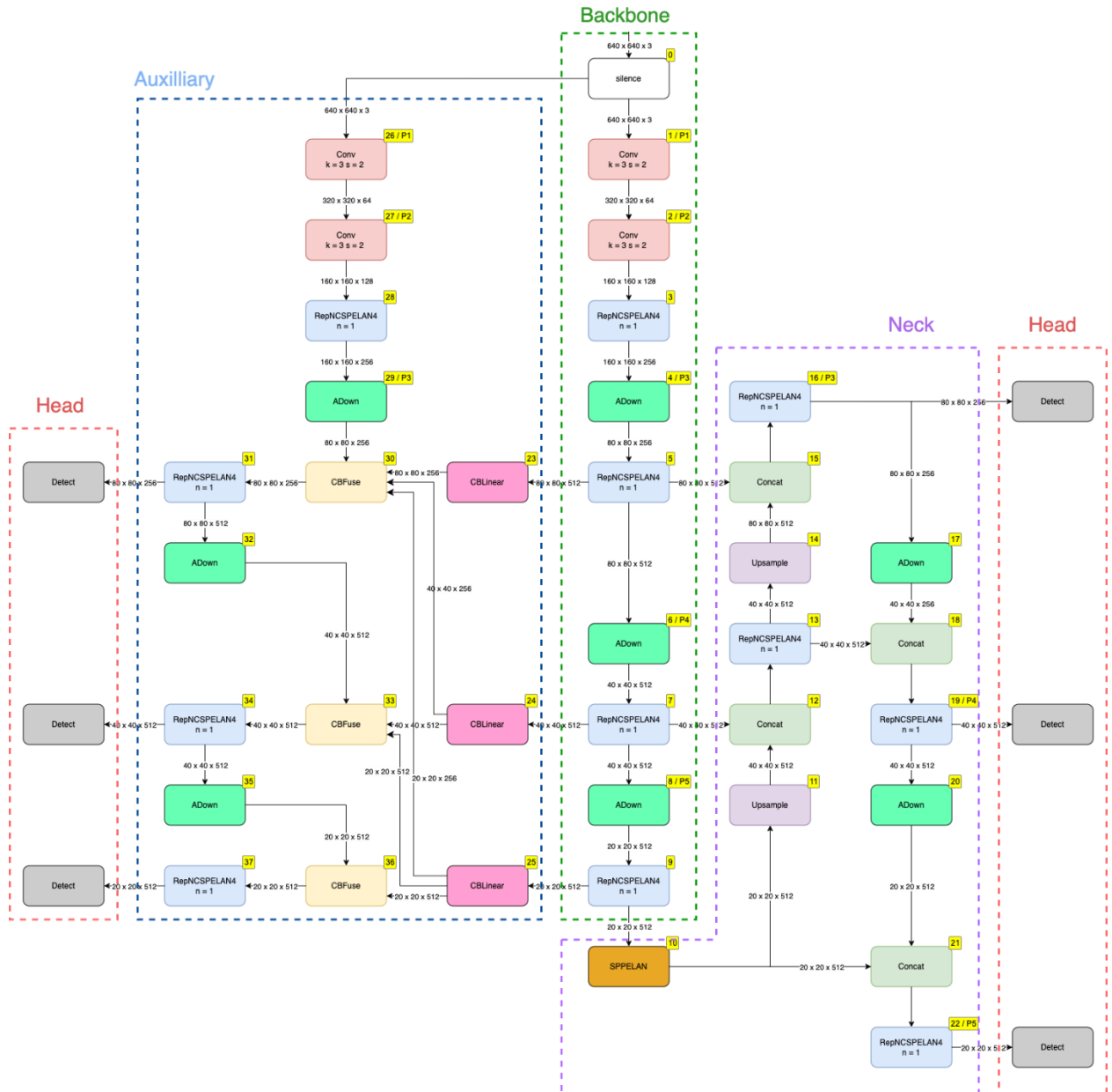


Рисунок 2.12 – Загальна архітектура YOLO v9

Надалі розглянемо складові частини архітектури обраної моделі для забезпечення ефективності подальшого розроблення нової технології для досягнення мети дисертаційного дослідження.

Розглянемо шар Conv2d (k, s, p, c). На схемі шар Conv2d (k, s, p, c) відповідає одному згортковому шару з двовимірним ядром (2d Convolution layer), що реалізує операцію згортки " * ", де параметри: k – kernel size, задає

розмір ядра згортки; s – stride, визначає крок згортки; p – padding, відповідає за заповнення елементів поза основною матрицею; c – channels, задає кількість каналів (карт ознак) згортки.

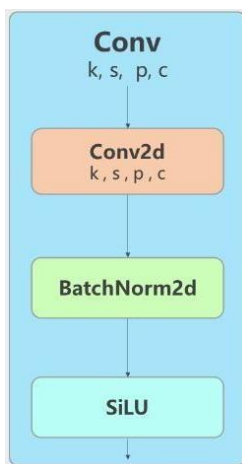


Рисунок 2.13 – Структура блоку Conv

Блок *Conv* (k, s, p, c) складається з трьох шарів (рисунок 2.13): шару Conv2d (k, s, p, c), шару нормалізації BatchNorm2d та шару активації SiLU. Пакетна нормалізація (Batch Normalization) має на меті зменшити внутрішній зсув коваріацій (covariate shift) і таким чином прискорити навчання глибоких нейронних мереж. Це досягається за допомогою кроку нормалізації, який фіксує середні значення та дисперсії вхідних даних шарів. Пакетна нормалізація також позитивно впливає на потік градієнта через мережу, зменшуючи залежність градієнтів від масштабу параметрів або їх початкових значень. Це дозволяє використовувати набагато вищі швидкості навчання без ризику дивергенції. Крім того, пакетна нормалізація впорядковує модель і зменшує потребу у шарах відсіву (Dropout layers). Застосовується шар пакетної нормалізації наступним чином для міні-пакетів (mini-batches) B :

$$\widehat{x}_l = \frac{x_i - M_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2.11)$$

$$y_i = \gamma \widehat{x}_l + \beta = BN_{\gamma\beta}(x_i) \quad (2.12)$$

де γ, β – змінюються під час навчання.

Сигмоїдні лінійні одиниці SiLU (Sigmoid Linear Units) є функціями активації для нейронних мереж (рисунок 2.14). Активация SiLU обчислюється за допомогою сигмоїдної функції, помноженої на її вхід:

$$SiLU(x) = x\sigma(x) \quad (2.13)$$

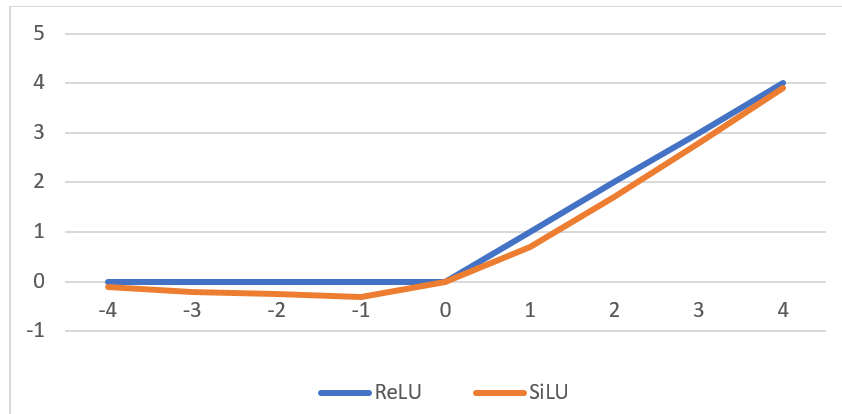


Рисунок 2.14 – Функція активації SiLU в порівнянні з ReLU

Блок *RepConvN*. Приділимо увагу процесу навчання (рисунок 2.15). Під час винесення висновку виконується перепараметризація (*fuse_convs*). У цей час *conv1*, *conv2* повторно параметризуються та перетворюються на операцію *conv*.

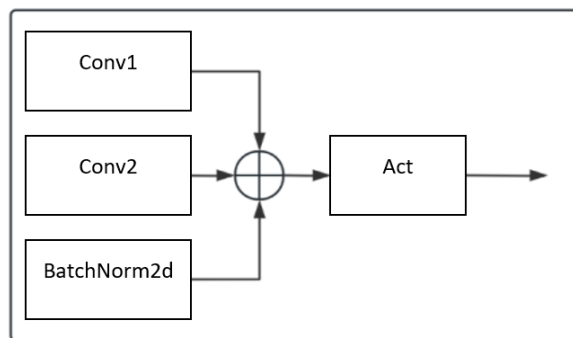


Рисунок 2.15 – Структура блоку RepConvN

Блок *RepNBottleneck*. Структура блоку RepNBottleneck зображена на рисунку 2.16.

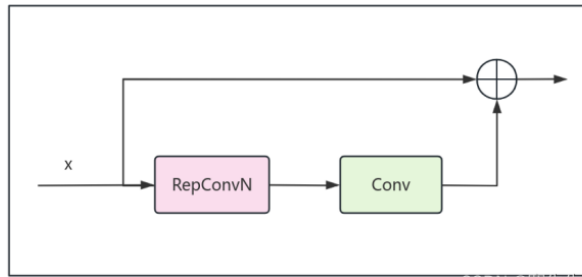


Рисунок 2.16 – Структура блоку RepNBottleneck

Блок *RepNCSP*. Структура блоку RepNCSP зображена на рисунку 2.17.

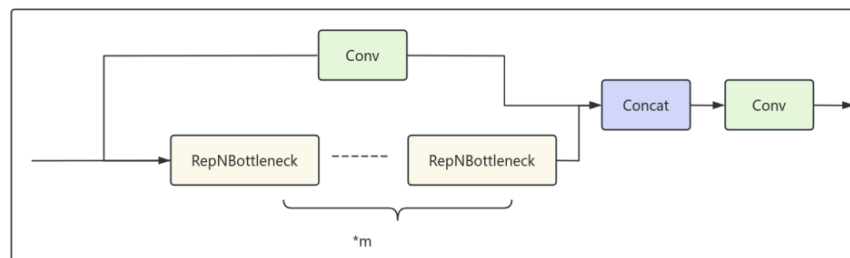


Рисунок 2.17 – Структура блоку RepNCSP

Блок *SPPELAN*. Цей модуль представляє підхід до агрегації шарів шляхом включення просторового пірамідального об'єднання (Spatial Pyramid Pooling SPP) в структуру ELAN.

ELAN (Efficient layer aggregation network). Ефективна мережа агрегації рівнів є стратегією проєктування мережевої структури, яка була вперше запропонована в статті «Проектування стратегій проєктування мережі через аналіз шляху» [40]. Проектування структури нейронної мережі в основному поділяється на два типи: проектування на основі шляху даних і проектування на основі градієнтного шляху. ELAN виконує відповідне проектування та оптимізацію на основі стратегії проектування градієнтного контуру, зосереджується на максимізації джерела градієнта та збагаченні градієнтного контуру. Насправді ця стратегія вже була реалізована в попередніх ResNet і CSPNet. Структура CSPNet, ELAN та GELAN зображено на рисунку 2.18.

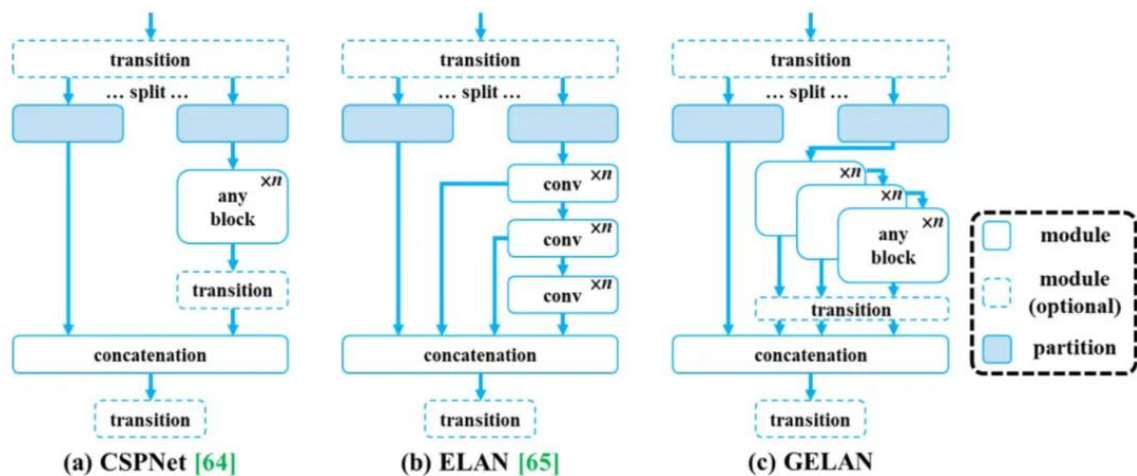


Рисунок 2.18 – Структура CSPNet, ELAN та GELAN

Блок *SPPELAN* починається зі згорткового шару, який коригує розміри каналу, після чого виконується серія операцій просторового об'єднання для захоплення різномасштабної контекстної інформації (multi-scale contextual information). Вихідні дані об'єднуються і проходять через ще один згортковий шар для консолідації ознак, оптимізуючи можливості мережі для детального вилучення ознак з різних просторових ієрархій.

Архітектура *GELAN* (General efficient layer aggregation network), анонсована разом з YOLOv9, об'єднує градієнтну ефективність CSPNet та орієнтовану на швидкість архітектуру ELAN в єдиний фреймворк, який підтримує ширший спектр обчислювальних блоків. Така гнучкість дозволяє YOLOv9 адаптуватися до різних обчислювальних середовищ і завдань, зберігаючи високу точність і швидкість.

Блок *RepNCSPPELAN4* (рисунок 2.19) являє собою вдосконалену версію CSP-ELAN, яка спрямована на подальшу оптимізацію процесу виділення ознак. Цей компонент розділяє вхідні дані з початкового згорткового шару на два шляхи, обробляє кожен з них через серію RepNCSP і згорткових шарів, а потім об'єднує їх назад. Ця двошляхова стратегія полегшує ефективний градієнтний потік і повторне використання функцій, значно підвищуючи ефективність навчання моделі та швидкість отримання висновків,

забезпечуючи глибину без обчислювального штрафу, який зазвичай асоціюється з підвищеною складністю.

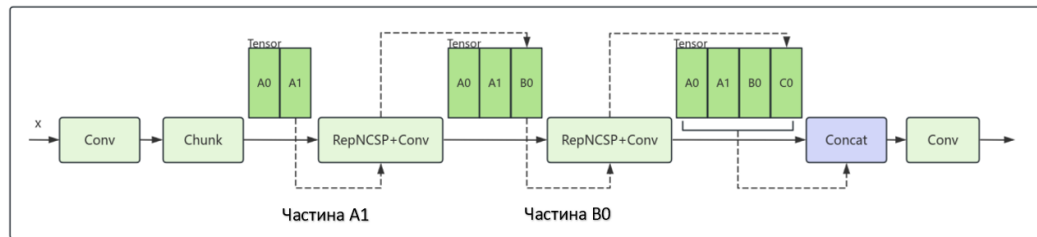


Рисунок 2.19 – Структура блоку RepNCSPELAN4

Блок CBLinear. Структура блоку CBLinear наведена на рисунку 2.20.

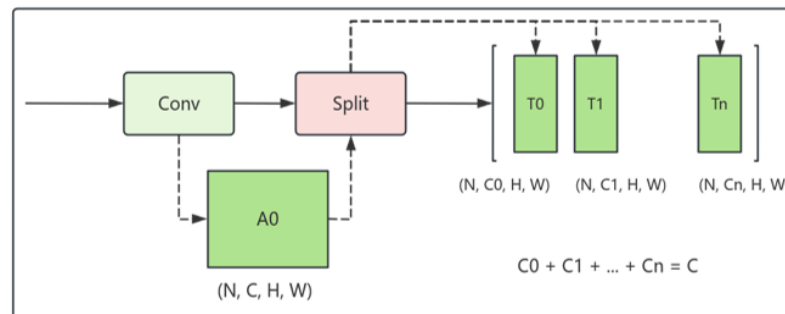


Рисунок 2.20 – Структура блоку CBLinear

Блок CBFuse. Вхід верхнього рівня CBFuse (рисунок 2.21) – CBLinear. Кожен виклик CUFuse є частиною вхідного тензора, а не агрегацією всього тензора.

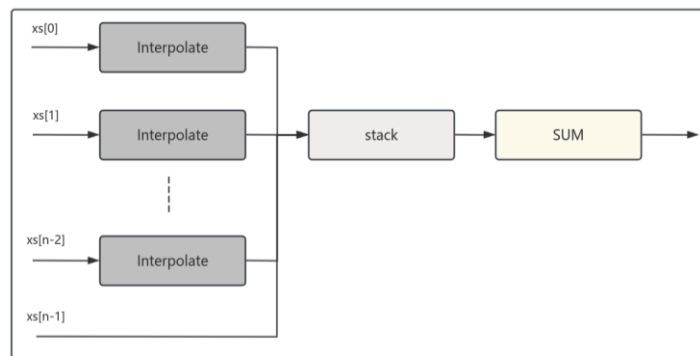


Рисунок 2.21 – Структура блоку CBFuse

2.2.3 Комплексна функція втрат

Функція втрат YOLO є комплексною, через багатомірність задачі розпізнавання об'єктів, та складається з трьох функцій втрат для кожної зі складових підзадач, кожна з яких в своїй природі основана на функції суми квадратичних похибок (sum-square error, SSE):

$$LossYolo = Lossconfidence + Losslocalization + Lossclassification,$$

де

Lossconfidence – функція втрат оцінки ймовірності присутності об'єкта (confidence loss);

Losslocalization – функція втрат локалізації (localization loss);

Lossclassification – функція втрат класифікації (classification loss).

Втрати оцінки ймовірності присутності об'єкта. Завдяки парадигмі YOLO з поділенням зображення на сітку блоків-клітинок, мережа генерує великий набір клітинок, які не містять жодного об'єкта, тому ці клітинки отримують нульову оцінку достовірності. Для більшості задач, ці порожні клітинки переважають над тими, що містять об'єкти, тим самим перекриваючи їх втрати та градієнт. Для вирішення цієї проблеми алгоритм YOLO зменшує втрати прогнозів з низькою достовірністю, встановлюючи гіперпараметр λ_{noobj} (параметр для керування процесом навчання) встановлюється 0.5.

Щоб гарантувати, що функція втрат достовірності сильно переважає функція втрат клітинок, які містять об'єкт (тобто навчання фокусується на клітинках, які містять об'єкт, а не порожніх), введено маску 1_{ij}^{obj} :

$$1_{ij}^{obj} = 1, \text{ якщо об'єкт існує в } i - \text{й клітинці та } j - \text{й рамка відповідає за його виявлення}$$

або

$$1_{ij}^{obj} = 0, \text{ інакше}$$

Аналогічно, щоб зменшити втрати для порожніх клітинок, встановлюється маска 1_{ij}^{noobj} :

$$1_{ij}^{noobj} = 1, \text{ якщо в клітинці немає об'єкта}$$

або

$$1_{ij}^{noobj} = 0, \text{ інакше.}$$

Це робить втрату довіри наступною:

$$Loss_{confidence} = \sum_i^{S^2} \sum_j^B 1_{ij}^{obj} (c_i - \hat{c}_j)^2 + \lambda_{noobj} \sum_i^{S^2} \sum_j^B 1_{ij}^{noobj} (c_i - \hat{c}_j)^2 \quad (2.14)$$

де

B – множина обмежувальних рамок однієї з S^2 клітинок

c_i та \hat{c}_j – реальна та передбачена оцінка наявності об'єкта в даній клітинці.

Втрати локалізації. Сума квадратичних похибок також вирівнює похибку у великих і малих обмежувальних рамках, що також не є ідеальним, оскільки невелике відхилення більш помітне у меншій рамці, ніж у більшій. Щоб вирішити цю проблему, YOLO прогнозує квадратний корінь з координат висоти та ширини замість безпосередньо висоти та ширини.

Крім того, функція втрат штрафує за помилку локалізації лише для передбаченої обмежувальної рамки, яка має найвищу IoU з реальною рамкою (з розмітки) у кожній клітинці сітки, щоб збільшити втрати для координат рамки і більше зосередитися на виявленні об'єкта, встановлено гіперпараметр $\lambda_{coord} = 5$.

Таким чином, функція втрат локалізації має наступний вигляд:

$$Loss_{confidence} = \sum_i^{S^2} \sum_j^B 1_{ij}^{obj} [(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2] + \lambda_{coord} \sum_i^{S^2} \sum_j^B 1_{ij}^{obj} [(\sqrt{h_i} - \sqrt{\hat{h}_j})^2 + (\sqrt{w_i} - \sqrt{\hat{w}_j})^2] \quad (2.15)$$

Втрати класифікації. Для функції втрат класифікації YOLO використовує суму квадратичних похибок для порівняння умовних ймовірностей $p_i(c)$ для всіх класів. Функція помилки враховувала помилку класифікації лише тоді, коли об'єкт присутній у клітинці сітки, тому використовується маска 1_i^{obj} :

$$Loss_{classification} = \sum_i^{s^2} 1_{ij}^{obj} \sum_c^C (p_i(c) - \widehat{p_i(c)})^2, \quad (2.16)$$

де C – множина всіх класів.

2.3 Опис власної технології YOLO v9 P

Згорткова нейронна мережа YOLO v9 є найсучаснішою технологією виявлення об'єктів і враховує різномасштабну природу об'єктів, використовуючи три шари виявлення масштабу для розміщення об'єктів різного масштабу. Однак зображення, отримані БпЛА, мають проблеми зі складним фоном і великою часткою дрібних об'єктів. Це призводить до того, що структура виявлення YOLOv9 не відповідає вимогам виявлення в сценаріях аерофотозйомки з БпЛА. Щоб пом'якшити вищезгадані проблеми, в цій роботі використовується YOLOv9 як базова модель і виконується оптимізація моделі з точки зору функції втрат та механізму уваги.

Основні ідеї стратегії покращення полягають у наступному:

- модуль WIoU v3 використовується як обмежувач регресійних втрат і включає в себе динамічний немонотонний механізм, що надає розумну стратегію розподілу градієнтного посилення, яка зменшує появу великих або шкідливих градієнтів від екстремальних вибірок; модуль WIoU v3 більше фокусується на вибірках звичайної якості, тим самим покращуючи як здатність моделі до узагальнення, так і загальну продуктивність;
- в опорну мережу вводиться динамічний механізм розрідженої уваги модуль ViFormer, який зменшує обчислення та споживання пам'яті,

заповнюючи більшість низькорелевантних областей на графі ознак, а потім звертає увагу на високорелевантні ознаки; модуль ViFormer покращує увагу моделі до ключової інформації у вхідних ознаках і оптимізує ефективність виявлення моделі;

– на основі FasterNet [45] запропоновано ефективний блок обробки ознак Perception FasterNet Block (PFNB), який використовує менше обчислень та звернень до пам'яті під час роботи; на основі PFNB було розроблено два нові шари виявлення.

Запропонована багатомасштабна мережа злиття ознак робить поверхневі ознаки і глибокі ознаки повністю взаємодоповнюючими, що ефективно покращує ефект виявлення моделі на малих об'єктах. Остаточну покращену мережеву модель YOLOv9-P і загальний фреймворк моделі показано на рисунку 2.22, де для позначення розмірів об'єктів використовується розподіл на типи об'єктів за розміром на зображенні: великі (80% зображення), середні (40% зображення), малі (20% зображення), X-малі (10% зображення) та XX-малі (5% зображення). Удосконалена модель змінює початкове 3-масштабне виявлення на 5-масштабне, що ефективно покращує загальну ефективність виявлення моделі, особливо для малих об'єктів.

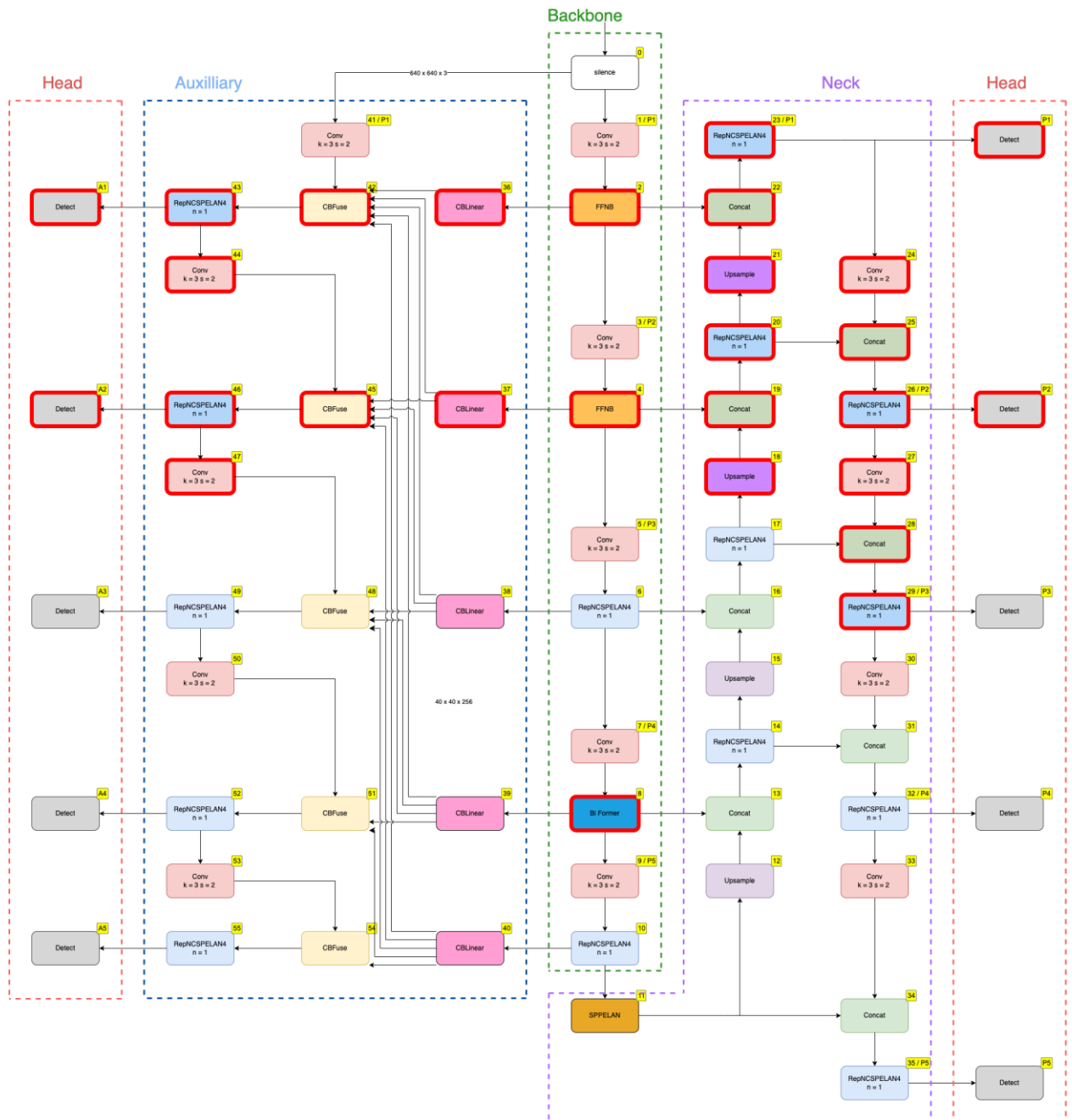


Рисунок 2.22 – Загальна архітектура запропонованої вдосконаленої моделі (з виділеними структурними блоками, які вбудовані в розроблене програмне рішення)

2.3.1 Покращення функції втрат

Задача виявлення об'єктів під час аерофотозйомки з БПЛА ускладнюється високою часткою дрібних об'єктів, тому прагматично

розроблена функція втрат може суттєво покращити ефективність виявлення моделі.

YOLOv9 використовує DFL та CIoU для обчислення регресійних втрат обмежувальної рамки, проте CIoU має такі недоліки: по-перше, CIoU не враховує баланс складних та простих вибірок. По-друге, CIoU використовує співвідношення сторін як один із штрафних множників функції втрат, і якщо співвідношення сторін реальної та прогнозованої рамки однакове, але значення ширини та висоти відрізняються, штрафний множник не може відобразити реальну різницю між цими двома рамками. По-третє, обчислення формули CIoU передбачає використання оберненої тригонометричної функції, що збільшує цикломатичну складність моделі. Формула CIoU наведена у рівнянні 2.17:

$$L_{CIoU} = 1 - IoU + \frac{p^2(b, b^{gt})}{(c_w)^2 + (c_h)^2} + \frac{4}{\pi^2} \left(\frac{w^{gt}}{h^{gt}} - \frac{w}{h} \right) \quad (2.17)$$

де

IoU – коефіцієнт перетину над об'єднанням, коефіцієнт перетину рамки передбачення і рамки реального об'єкта;

$p^2(b, b^{gt})$ – евклідова відстань між центроїдами реальної рамки і прогнозованої;

h та w – висота і ширина прогнозованої рамки;

h^{gt} і w^{gt} – висота і ширина реальної рамки;

c_h та c_w – висота і ширина мінімальної охоплюючої рамки, утвореної рамкою прогнозу і реальною рамкою.

Деякі з параметрів, що входять до рівняння 2.17, показано на рисунку 2.23.

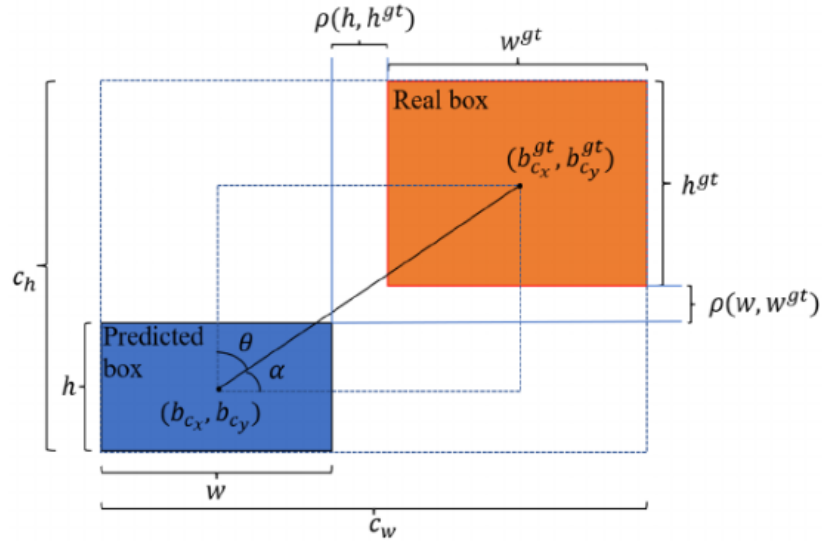


Рисунок 2.23 – Схематична діаграма параметрів функції втрати

EIoU [41] вдосконалює CIoU, розглядаючи довжину та ширину окремо як штрафні члени, що відображають різницю в ширині та висоті між реальною та прогнозованою рамками, що є більш обґрунтованим порівняно зі штрафним членом CIoU. Формула для обчислення EIoU має вигляд:

$$L_{EIou} = 1 - IoU + \frac{p^2(b, b^{gt})}{(c_w)^2 + (c_h)^2} + \frac{p^2(w, w^{gt})}{(c_w)^2} + \frac{p^2(h, h^{gt})}{(c_h)^2} \quad (2.18)$$

Деякі з параметрів, що входять до рівняння (2.18), показані на рисунку 2.23. $p^2(w, w^{gt})$ та $p^2(h, h^{gt})$ позначають евклідову відстань по ширині і евклідову відстань по висоті між реальною рамкою і рамкою прогнозу, відповідно.

SIoU [42] вперше вводить кут між прогнозованою та реальною областями як штрафний фактор. По-перше, виходячи з величини кута (як на рисунку 2.23, θ та α) між прогнозованою та реальною областями, прогнозована область швидко рухається до найближчої осі, а потім регресує до реальної області. SIoU зменшує ступені свободи регресії і прискорює збіжність моделі.

У той час як кілька основних функцій втрат, представлених вище, використовують статичний механізм фокусування, WIoU не тільки враховує аспект, міжцентрову відстань і площу перекриття, але також вводить

динамічний немонотонний механізм фокусування. WIoU застосовує розумну стратегію розподілу градієнта посилення для оцінки якості якірної коробки. Автори Тонг та ін. [43] запропонували три версії WIoU: v1 було розроблено з урахуванням прогнозованих втрат на основі уваги, у v2 і v3 додали коефіцієнти фокусування. WIoUv1 вводить відстань як метрику уваги. Коли об'єктний блок і передбачуваний блок перекриваються в межах певного діапазону, штраф за зменшення геометричної метрики дозволяє моделі отримати кращу здатність до узагальнення. Формула для розрахунку $L_{WIoU\ v1}$ наведена в рівняннях 2.19 – 2.21:

$$L_{WIoU\ v1} = R_{WIoU} \times L_{IoU} \quad (2.19)$$

$$R_{WIoU} = \exp\left(\frac{(b_{cx}^{gt} - b_{cx})^2 + (b_{cy}^{gt} - b_{cy})^2}{(c_w^2 + c_h^2)}\right) \quad (2.20)$$

$$L_{IoU} = 1 - IoU \quad (2.21)$$

Функція WIoUv2 застосовується до WIoUv1 шляхом побудови монотонного коефіцієнта фокусування L^*IoU , який ефективно зменшує вагу простих прикладів у значенні втрат. Однак, враховуючи, що L^*IoU зменшується в міру того, як L_{IoU} зменшується під час навчання моделі, що призводить до сповільнення збіжності, для нормалізації L^*IoU вводиться середнє значення L_{IoU} . Формула для обчислення $L_{WIoU\ v2}$ має вигляд:

$$L_{WIoUv2} = \left(\frac{L^*IoU}{L_{IoU}}\right)^\gamma \times L_{WIoUv1,\gamma} > 0 \quad (2.22)$$

WIoUv3 визначає відхилення β для вимірювання якості опорного кадру, буде немонотонний коефіцієнт фокусування r на основі β і застосовує r до WIoUv1. Мале значення β вказує на високу якість опорного кадру, і йому присвоюється менший коефіцієнт підсилення, що зменшує вагу високоякісних опорних кадрів у більшій функції втрат. Велике значення β вказує на середню якість анкерного блоку, і йому присвоюється невеликий коефіцієнт підсилення градієнта, що зменшує шкідливі градієнти, які генеруються низькоякісними анкерними блоками. WIoUv3 використовує розумну стратегію розподілу

градієнтного підсилення для динамічної оптимізації ваги високоякісних і неякісних анкерних блоків у втратах, що змушує модель фокусуватися на зразках середньої якості і покращує загальну продуктивність моделі. Формули $WIoUv3$ наведено в рівняннях (2.23)-(2.25). δ і α є гіперпараметрами, які можна налаштувати для різних моделей

$$L_{WIoUv3} = r \times L_{WIoUv1} \quad (2.23)$$

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \quad (2.24)$$

$$\beta = \frac{L_{IoU}^*}{L_{IoU}} \in [0, +\infty) \quad (2.25)$$

Порівнюючи кілька основних функцій втрат, було вирішено використовувати функцію $WIoUv3$ для обчислення регресійних втрат в об'єктній області. З одного боку, $WIoUv3$ враховує деякі переваги $EIoU$ та $SIoU$, що відповідає концепції дизайну оптимальної функції втрат. З іншого боку, $WIoU v3$ використовує динамічний немонотонний механізм для оцінки якості анкерних блоків, що змушує модель більше фокусуватися на анкерних блоках звичайної якості і покращує здатність моделі локалізувати об'єкти. У задачі виявлення об'єктів під час аерофотозйомки з БПЛА висока частка дрібних об'єктів збільшує складність виявлення, і $WIoU v3$ може динамічно оптимізувати вагу втрат дрібних об'єктів, щоб покращити ефективність виявлення моделі.

2.3.2 Ефективний механізм уваги

Ефективний механізм уваги дозволяє будувати надійні та потужні моделі на основі даних, що робить моделі більш гнучкими при роботі зі складними та великими даними. Механізм уваги працює наступним чином: спочатку $[x_1, x_2, x_3, \dots, x_T]$ отримується шляхом кодування послідовності вхідних даних $[a_1, a_2, a_3, \dots, a_T]$. Потім матриці запитів Q , ключів K та значень V , отримуються за допомогою матриць лінійних перетворень W^Q , W^K та

W^V відповідно. Обчислюється добуток між запитом і відповідним ключем, потім нормалізується і множиться на матрицю V для отримання зваженої суми. Величина d_K позначає вимірність матриці K і $\sqrt{d_K}$ вводиться для того, щоб запобігти зникненню градієнта результату. Обчислення механізму уваги можна виконати за формулою:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (2.26)$$

Однак звичайний механізм уваги має недоліки, пов'язані з високою обчислювальною складністю та великим використанням пам'яті. Моделі виявлення, що використовуються на бортах БпЛА, обмежені в ресурсах, і якщо ввести звичайний модуль уваги безпосередньо в модель, він займе більшу частину платформи і знизить швидкість інференції моделі. Щоб пом'якшити проблеми з кількістю обчислень та необхідним об'ємом пам'яті, дослідники запропонували зменшити споживання ресурсів, замінивши глобальні запити розрідженими запитами, які фокусуються лише на деяких парах ключ-значення. Існує багато суміжних робіт, заснованих на цій дослідницькій ідеї, таких як локальна увага, деформована увага та експансивна увага, але всі вони є ручним створенням статичних патернів і незалежної від змісту розрідженості. Щоб вирішити ці проблеми, автори Чжу та ін. [44] запропонували новий механізм динамічної розрідженої уваги: дворівневу маршрутизуючу увагу, процес роботи якої показано на рисунку 2.24.

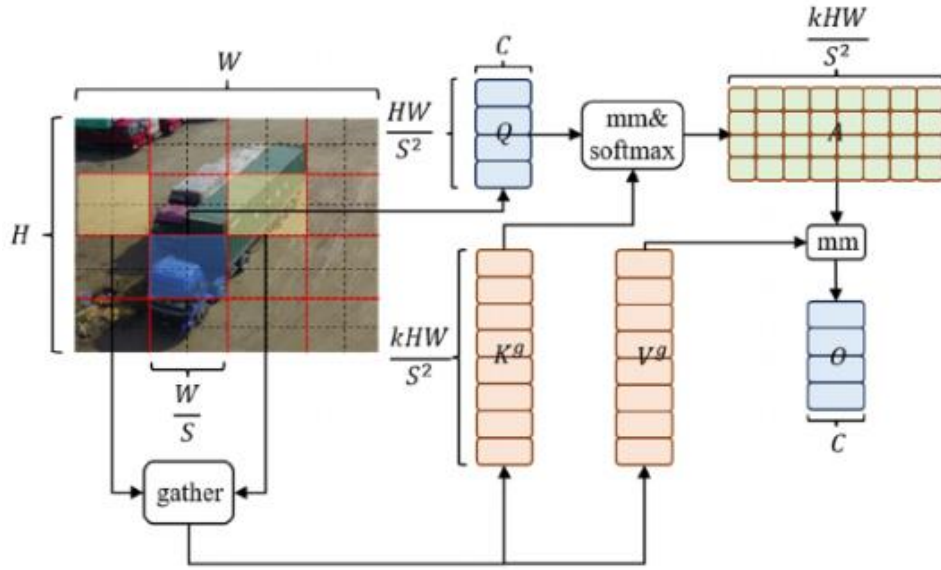


Рисунок 2.24 – Структура механізму дворівневої маршрутизуючої уваги

На рисунку 2.24 видно, що вхідна карта ознак $x \in R^{H \times W \times C}$ спочатку розбита на $S \times S$ підмножин, кожна з яких містить $\frac{HW}{S^2}$ векторів ознак. Проводиться заміна форми X так, що результатом є

$$X^r \in R^{S^2 \times \frac{HW}{S^2} \times C}.$$

Потім вектори ознак лінійно трансформуються для отримання матриць Q , K та V :

$$Q = X^r W^Q \quad (2.27)$$

$$K = X^r W^K \quad (2.28)$$

$$V = X^r W^V \quad (2.29)$$

Відхилення уваги від регіону до регіону отримується шляхом побудови орієнтованого графа. Цей граф використовується для знаходження регіонів, які пов'язані з поточним. Конкретний процес реалізації виглядає наступним чином: матриці Q та V для кожного регіону обробляються шляхом усереднення по регіонах для отримання значення Q^r та $K^r \in R^{S^2 \times C}$ на рівні регіону. Потім обчислюється добуток Q^r та K^r для отримання матриці

суміжності $A^r \in R^{S^2 \times S^2}$, яка використовується для вимірювання міжрегіональної кореляції:

$$A^r = Q^r (K^r)^T \quad (2.30)$$

Далі A^r обрізається. Найменш релевантні токени в A^r відсіюються на низькодеталізованому рівні, а верхні k найбільш релевантних областей в A^r залишаються для отримання матриці індексів маршрутизації, $I^r \in N^{S^2 \times k}$:

$$I^r = \text{topkIndex}(A^r) \quad (2.31)$$

Згодом на високодеталізованому рівні використовується механізм уваги від токена до токена. Для запитів у i -му регіоні цей механізм уваги фокусується лише на регіонах маршрутизації, де індексуються $I_{(i,1)}^r, I_{(i,2)}^r, \dots, I_{(i,k)}^r$ і збирає всі тензори K і V в цих регіонах для отримання K^g і V^g :

$$K^g = \text{gather}(K, I^r) \quad (2.32)$$

$$V^g = \text{gather}(V, I^r) \quad (2.33)$$

Наприкінці алгоритму, отримані K^g та V^g обробляються з механізмом уваги і додається локальний член покращення контексту від матриці V (local context enhancement term, LCE), щоб отримати вихідний тензор O , формула для обчислення якого показана нижче:

$$O = \text{Attention}(Q, K^g, V^g) + \text{LCE}(V) \quad (2.34)$$

Блок ViFormer розроблено на основі механізму дворівневої маршрутизуючої уваги, як показано на рисунку 2.25.

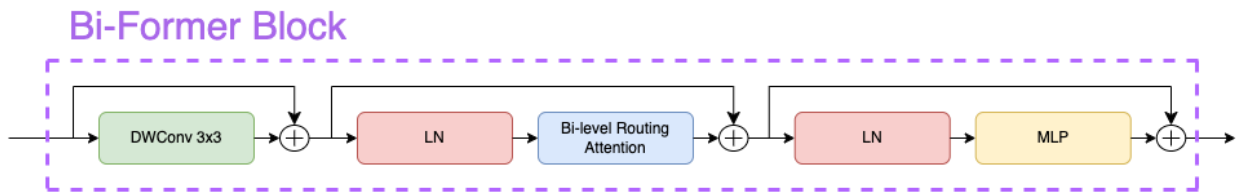


Рисунок 2.25 – Структура блоку ViFormer

Блок DWConv у цьому блоці позначає глибоку відокремлювану згортку, яка може зменшити кількість параметрів та обчислень в моделі. LN позначає

обробку нормалізації шару, яка може прискорити навчання та покращити здатність моделі до узагальнення. MLP позначає багатошаровий персептрон, який додатково обробляє та налаштовує ваги механізму уваги, щоб посилити увагу моделі до різних ознак. Символ додавання на рисунку вказує на з'єднання двох векторів ознак.

У дисертаційному дослідженні блок BiFormer введено до хребта моделі YOLOv9. З одного боку, BiFormer може враховувати обмеженість обчислювальних потужностей та ресурсів пам'яті апаратної платформи БПЛА. З іншого боку, механізм динамічної уваги цього блоку дозволяє покращити увагу моделі до критично важливої інформації про об'єкт та оптимізувати ефективність виявлення моделі. Для повного використання ефективного механізму уваги в цьому блоці використовується блок BiFormer між 4:Conv та 6:Conv опорної мережі моделі, замінивши оригінальний блок RepNCSPELAN4.

2.3.3 Багатомасштабна мережа об'єднання ознак

Погане виявлення малих об'єктів є однією з проблем у задачах виявлення об'єктів в контексті аерофотозйомки з БПЛА. У багатьох існуючих роботах для зменшення частоти пропусків малих об'єктів до моделі додають шкали виявлення, що є ефективним методом покращення. Однак такий підхід може ускладнювати структуру моделі та збільшувати споживання обчислювальних ресурсів і ресурсів для зберігання даних. Для вирішення цієї проблеми в роботі запропоновано блок обробки ознак, який називається PFNB, та розроблено багатомасштабну мережу злиття ознак на основі цього блоку. Точність виявлення малих об'єктів значно покращується при зменшенні надмірного споживання ресурсів.

Завдання виявлення об'єктів для платформ БПЛА обмежені обчислювальними ресурсами, тому шукаються моделі з простою структурою,

низькою затримкою і високою пропускнуою здатністю. Деякі класичні легкі мережі, такі як MobileNet, ShuffleNet і GhostNet, використовують глибоку згортку або групову згортку для вилучення просторових ознак зображень. Глибока згортка зменшує вхідні розміри ознак шляхом згортки вхідних зображень, згрупованих за розмірами ознак, зменшуючи кількість параметрів, зберігаючи при цьому інформацію про ознаки в основному незмінною. Групову згортку можна розглядати як розріджену форму традиційної згортки, коли вхідні канали згортаються один за одним, що може бути використано для зменшення параметрів моделі та досягнення мети легких моделей. Більшість таких полегшених моделей зосереджені на зменшенні кількості операцій над точками (FLOP), і дуже мало відповідних робіт розглядають низьку кількість операцій над точками за секунду (FLOPS) моделі. Однак, зменшення параметрів моделі не призводить до збільшення швидкості обчислень моделі. Тому деякі роботи з використанням глибокої або групової згортки в спробі спроектувати легкі та швидкі блоки нейронної мережі в деяких випадках не прискорюють роботу моделі, а навіть погіршують затримку.

Для вхідної ознаки розміром $h \times w \times c$, необхідна кількість FLOP при використанні регулярної згортки розміром $k \times k$ обчислюється за формулою.

$$FLOPs_{Conv} = h * w * k^2 * c^2 \quad (2.35)$$

Ядро глибокої згортки виконує операцію ковзання над простором вхідного каналу, щоб отримати характеристики вихідного каналу, і обчислює FLOP для глибокої згортки:

$$FLOPs_{DWConv} = h * w * k^2 * c \quad (2.36)$$

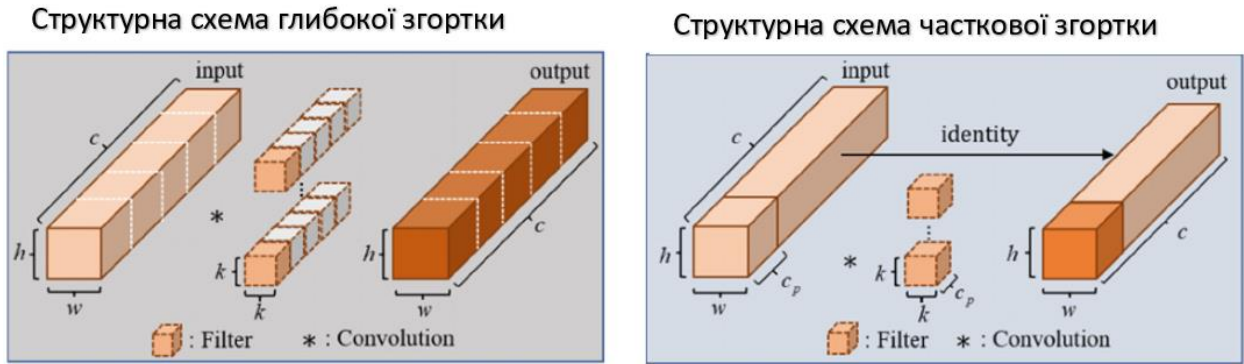


Рисунок 2.26 – Порівняння DWConv та PConv

Популярна глибинна згортка, процес обчислення якої показано на рисунку 2.26, ефективно зменшує параметри моделі. Але на практиці глибинна згортка потребує додаткової точкової згортки або інших обчислювальних витрат, щоб компенсувати зниження точності після операції згортки, що призводить до додаткових витрат на доступ до пам'яті та збільшує затримку. Для виправлення цих недоліків, автори Чен та ін. [45] запропонували часткову згортку (PConv). PConv використовує звичайну згортку для виконання операції згортки над деякими неперервними елементами вхідного каналу, а решта елементів обробляється за допомогою тотожного відображення, залишаючи канал незмінним. Виконаємо обчислення згортки для першого послідовного елемента з номером каналу c , підставивши вхідні елементи, як показано на рисунку 2.27, і виведемо формулу для обчислення FLOPs PConv:

$$FLOPs_{PConv} = h * w * k^2 * c_p^2 \quad (2.37)$$

Якщо c_p становить 1/4 від кількості каналів вхідних ознак, то $FLOPs_{PConv}$ становить лише 1/16 від звичайної згортки. Ця згортка зменшує кількість звернень до пам'яті, зменшуючи при цьому параметри, і ефективно витягує просторові особливості вхідної інформації. Автори Чен та ін. [45] запропонували блок FasterNet на основі PConv, який являє собою модуль, що

складається з шару PConv і двох згорткових шарів 1×1 , з'єднаних послідовно, як показано на рисунку 2.27.

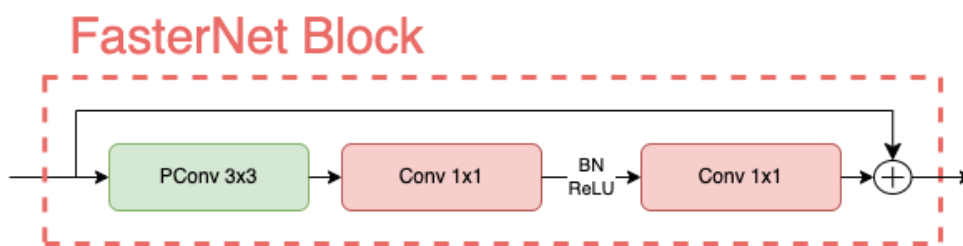


Рисунок 2.27 – Структура блоку FasterNet (FNB)

Символ додавання на рисунку 2.27 вказує на з'єднання двох векторів ознак. Надмірне використання шарів нормалізації та активації може призвести до зменшення різноманітності ознак, що може вплинути на продуктивність моделі. Тому в блоці FasterNet шари нормалізації та активації використовуються тільки після другого шару згортки. Блок FasterNet має просту структуру і невелику кількість параметрів для швидшої роботи.

У дисертаційному дослідженні переглянуто структуру блоку FasterNet. У цьому блоці використовується шар згортки 1×1 , що дозволяє зменшити кількість параметрів, пришвидшити навчання та підвищити здатність моделі до нелінійної підгонки. Однак область сприйнятливості згортки 1×1 є відносно невеликою і їй не вистачає для набуття глобальних особливостей. Також враховано, що блок FasterNet використовує лише одне коротке з'єднання, а вхідні ознаки згортаються через три шари, що може призвести до деградації мережі та зникнення ознак у міру поглиблення моделі в глибину. Для вирішення вищезазначених проблем у дисертаційному дослідженні запропоновано PFNB на основі блоку FasterNet, структура якого показана на рисунку 2.28.

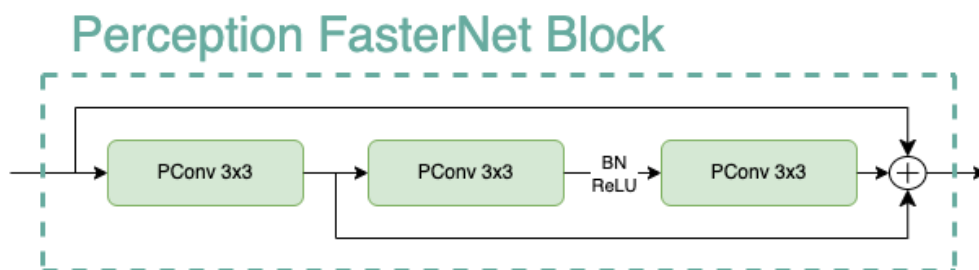


Рисунок 2.28 – Структура Perception FasterNet Block (PFNB)

По-перше, PConv використовується для заміни двох шарів згортки 1×1 у блоці FasterNet, що покращує сприйнятливості поле, одночасно роблячи вихідний модуль швидшим та ефективнішим. По-друге, залишкова конкатенація додається до двох останніх шарів згортки в блоці, щоб збагатити характеристики вихідної інформації, зменшити втрату неефективних характеристик і оптимізувати ефективність виявлення моделі.

Більшість сучасних моделей виявлення об'єктів використовують згорткові нейронні мережі для вилучення ознак об'єктів. Зі збільшенням кількості згорток семантична інформація вхідних ознак поступово стає багатшою, але детальних ознак стає все менше і менше, що є однією з головних причин низької точності виявлення багатьох моделей виявлення малих об'єктів. Хоча YOLOv8 використовує багатомасштабний метод виявлення, він все ще не може задовольнити потреби виявлення в сценаріях аерофотозйомки БПЛА, що призводить до незадовільної точності виявлення моделі для малих об'єктів. Точки входу блоків детекції YOLO v8 зображено на рисунку 2.29.

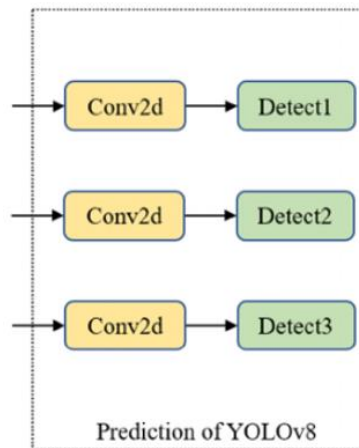


Рисунок 2.29 – Точки входу блоків детекції YOLO v9

Щоб підвищити точність виявлення малих об'єктів і врахувати обмежені ресурси платформи, в цій роботі використовується ефективний PFNB для проєктування мережі злиття ознак. У дисертаційному дослідженні додано дві нові шкали виявлення до оригінальних трьох шкал виявлення YOLOv8 і об'єднали поверхневу інформацію B1 і B2, які є більш багатими на інформацію про місцезнаходження. PFNB додається як блок обробки ознак між B1 і B2 магістральної мережі, а оригінальний блок C2f між B2 і B3 замінюється цим блоком. Впровадження PFNB дозволяє зменшити споживання ресурсів, спричинене різномасштабним злиттям ознак. Удосконалена модель досягає п'ятимасштабного виявлення, як показано на рисунку 2.30, що ефективно покращує ефективність виявлення моделі.

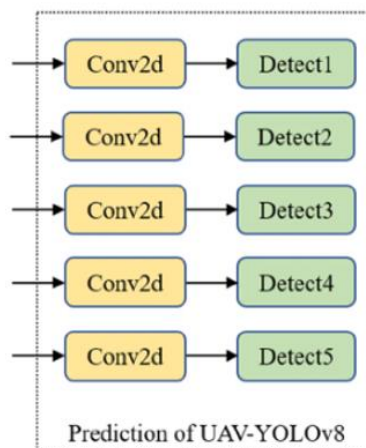


Рисунок 2.30 – Точки входу блоків детекції YOLO v9 P

2.4 Алгоритми препроцесингу зображень

Як було зазначено в розділі 1.1.3 даної роботи, на якість зображення впливає багато факторів, що потенційно можуть погіршити його якість та результати подальшого аналізу даних отриманих з зображення, внаслідок чого, розпізнавання на відстеження об'єктів погіршиться. Серед цих проблем є ті, що зустрічаються найчастіше та можуть бути вирішенні алгоритмічно, як задачі:

- видалення шуму;
- видалення ефекту розмиття;
- збільшення контрастності.

2.4.1 Видалення шуму

Видалення шуму полягає у зменшенні шумів у зображенні (у більшості випадків, неможливо повністю прибрати шум). Основні джерела шуму в цифрових зображеннях виникають під час зйомки (замала кількість зібраних фотонів, температура сенсора) або під час передачі (відлуння та атмосферні спотворення при бездротовому зв'язку). При передачі зображення в аналоговому вигляді шум з'являється від мінімальних перешкод в каналі передачі.

Шум за своєю природою є випадковим явищем, він моделюється щільністю ймовірності, яка відображає розподіл інтенсивності шуму. У подальшому розгляді різних моделей будуть використовуватися наступні позначення: y – зашумлене зображення, b – шум, x – зображення без шуму.

Виділяють основні типи шумів, які використовуються у спостереженнях з різними умовами та вимогами.

Наприклад, вплив шуму на зображення (рисунок 2.31) для адитивного гаусівського шуму все зображення зазнає однакового впливу шуму, для

пуассонівського шуму світліші частини є більш шумними, ніж темні, для імпульсного шуму змінюється лише кілька пікселів, які замінюються чорними або білими пікселями.

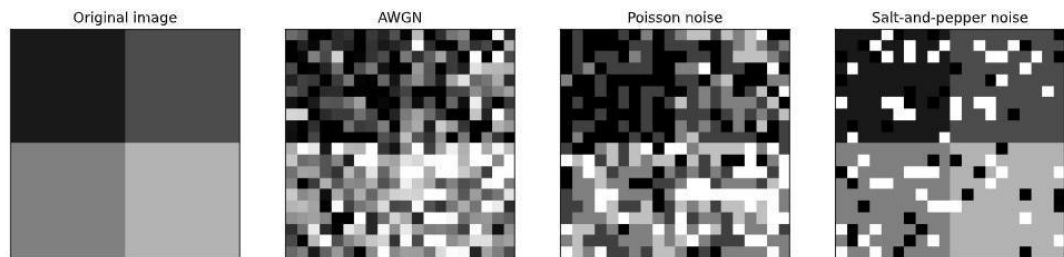


Рисунок 2.31 – Вплив різних видів шуму на зображення

Для позбавлення від шуму можуть використовувати медіанний фільтр, який не є фільтром за визначенням, оскільки він не дотримується властивості лінійності, і не підходить під визначення згортки. З існуючих методів позбавлення шуму можна виділити методи TV-регуляризації.

Задача отримання зображення без шуму \hat{x} , близького до спостереження y , призводить до критерію «відповідності даних», який вимірює різницю між y та x . Класичним вибором для вимірювання цієї різниці є критерій найменших квадратів:

$$E(x, y) = \sum_{i,j} (x_{ij} - y_{ij})^2 \quad (2.38)$$

Чим ближче зображення x до y , тим менше $E(x, y)$. Задача отримати зображення з малими варіаціями інтенсивності призводить до появи критерію «регуляризації», який вимірює різницю між сусідніми пікселями. Простим вибором є мінімізація «сумарної варіації» (total variation):

$$R(x) = \sum_{i,j} |x_{ij} - x_{i+1,j}| + \sum_{i,j} |x_{ij} - x_{i,j+1}| \quad (2.39)$$

Мета TV-регуляризації полягає у тому, щоб знайти зображення x яке мінімізує як підбір даних, так і регуляризацію. Математично, потрібно шукати зображення x яке мінімізує $E(x, y) + \lambda R(x)$, де λ – це гіперпараметр

регуляризації, який використовується для налаштування компромісу між двома критеріями. Отже принци TV- регуляризації:

$$\hat{x} = E(x, y) + \lambda R(x) \quad (2.40)$$

Це призводить до проблеми обчислювальної складності, яка в програмному рішенні дисертаційної роботи вирішується за рахунок вбудованих бібліотек з реалізованими методами оптимізації.

2.4.2 Видалення ефекту розмиття

Розмиття (blur) є проблемою, яка пов'язана з втратою чіткості (sharpness) та різноманіття інформації на зображенні. Дане явище з'являється з різних причин, ініційованих як технічною недосконалістю апаратури, так і фізичними впливами під час створення або передачі зображення.

Загальне формулювання моделі має вигляд:

$$I_B = K * I_S + N \quad (2.41)$$

де

I_B – розмите зображення (blurred image),

K – ядро розмиття (PSF, Point spread function),

I_S – чітке приховане зображення, позначає операцію згортки,

N – додатковий шум.

Розмиття класифікується за ядром згортки, яке є джерелом розмиття при згортковій моделі, та власне природою причини при якій дане розмиття утворилося на зображенні. Найбільш популярною є класифікація на наступні чотири основні типи:

– розмиття від невірного фокусу (defocus blur, out-of-focus, disk blur) спостерігається, коли камера некоректно фокусується на об'єкті;

– розмиття від обробки (Processing Blur) виникає при застосуванні різних методів препроцесингу зображень для уникнення локальних дефектів, які можуть вплинути на якість зображення (наприклад, Гаусівське розмиття);

– розмиття внаслідок руху (Motion blur), яке ділиться на розмиття внаслідок руху сцени (коли відносно всієї сцени зображення камера здійснює рух, наприклад при розвороті камери) та розмиття внаслідок руху певних об'єктів (коли на відносно нерухомій сцені розмиттю піддаються лише рухомі об'єкти), на що додатково впливає характер руху (наприклад нерівномірність швидкості за час затримки камери, напрямок руху);

– змішане розмиття (mixed blur), ядро згортки якого є зваженою сумою ядер типів розмиття, з якого він складається.

Для видалення розмиття на зображенні використовують групи методів:

1) Методи видалення розмиття без використання нейронних мереж.

Для зменшення наслідків розмиття від невірного фокусу та від обробки зображення можна використовувати згортки для підвищення чіткості зображення та виділення контурів, наприклад ядро розміром 3×3 . Але даний метод частково працює лише для частини типів розмиття, при завчасно відомому ядру розмиття.

Фільтрація Вейнера (Wiener Filtering) є методом видалення розмиття та шуму, який заснований на статистичних принципах для покращення якості зображень.

Алгоритм Richardson-Lucy є ітеративним методом деконволюції, розроблений для відновлення зображень і головним недоліком даного якого є його повільність.

Загалом всі задачі видалення розмиття діляться на сліпі (blind) та несліпі (non-blind), в залежності від того чи попередньо відомо характер розмиття. Вищеописані методи використовують для несліпих задач, тому що потребують інформацію про розмиття перед обробкою зображення.

2) Використання згорткових нейронних мереж.

Враховуючи те, що ядро згортки не є сталим для кожної частини зображення, методи засновані на використанні згорткових нейронних мереж спочатку визначають ядро розмиття, а потім, використовуючи фільтрацію, видаляють розмиття з зображення. На жаль, цей підхід не працює для всіх типів розмиття й не є універсальним рішенням. Цю проблему вирішує використання згорткових нейронних мереж, як для передбачення ядра, так і для кінцевого видалення розмиття.

3) Генеративні змагальні мережі (Generative Adversarial Networks, GAN).

Ідея GAN полягає у визначенні «гри» між двома конкуруючими мережами: дискримінатором та генератором. Генератор отримує на вхід шум і генерує вибірку. Дискримінатор отримує реальну та згенеровану вибірку і намагається їх розрізнити. Метою генератора є обманути дискримінатор, формуючи переконливі для сприйняття зразки, які неможливо відрізнити від справжніх. З теоретичної точки зору, гра між генератором G та дискримінатором D є мінімаксною задачею:

$$[\log(D(x))] - [\log(1 - D(\tilde{x}))] \quad (2.42)$$

де

P_r – розподіл даних,

P_g – розподіл моделі.

GAN відомі своєю здатністю генерувати вибірки хорошої якості сприйняття, однак, базової версії страждають від багатьох проблем, таких як зникаючі градієнти. Дану проблему частково вирішує використання відстані Earth-Mover (також звану Wasserstein-1) для побудови WGAN (рисунок 2.32). Мережа WGAN з додаванням додаткової регуляриції дозволяє стабільно тренувати широкий спектр архітектур GAN майже без гіперпараметричного налаштування.

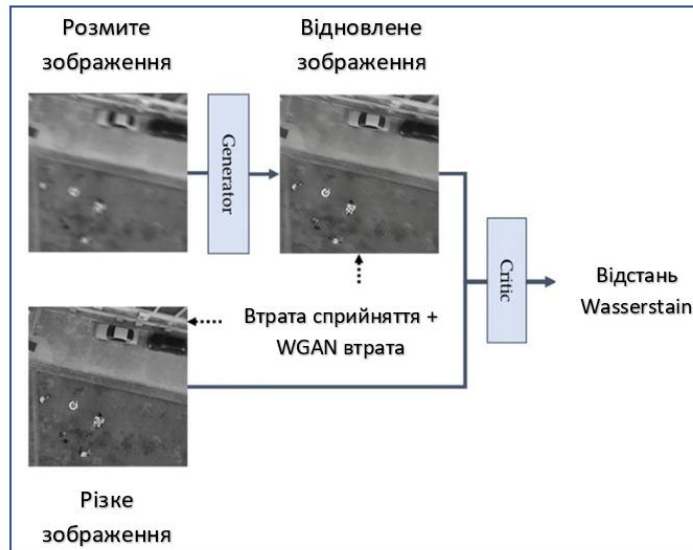


Рисунок 2.32 – Принцип роботи WGAN

DeblurGAN. DeblurGAN є представниками глибокої нейронної мережі, яка є розробленою для видалення розмиття з зображень на основі підходу GAN з використанням відстані Васеріяна [46]. Генеративна частина мережі відповідає за створення чіткого зображення з розмитого.

Архітектура нейронної мережі генератора DeblurGAN (рисунок 2.28) складається з двох згорткових блоків зі зміщенням 0.5, послідовності з дев'яти ResBlock (Residual Blocks) блоків та двох транспонованих згорткових блоків.

Кожен блок ResBlock в свою чергу складається з згорткового шару, шару нормалізації (Batch Normalization) та функції активації ReLU. Шар регуляризації Dropout з ймовірністю 0.5 знаходиться після згорткового шару кожного ResBlock.

Додатково використовується зв'язок глобального пропуску (global skip connection). Згорткова нейронна мережа генератора застосовує залишкову корекцію I_R до розмитого зображення I_B для створення чіткого зображення I_S , тобто:

$$I_S = I_R + I_B \quad (2.43)$$

- засліплення кадру,
- тотальна перевага певного спектру на зображенні,
- наявність фізичного фільтру.

Найвища контрастність досягається при розподілі кольорів та тонів на зображенні, що рівномірно займає весь видимий спектр. Задача збільшення контрастності полягає в зміні цього розподілу на зображенні.

Розглянемо у якості обраного методу збільшення контрастності метод вирівнювання гістограми (Histogram Equalization).

Цей метод підвищує загальну контрастність зображення, особливо коли воно представлено вузьким діапазоном значень інтенсивності. Завдяки цьому коригуванню інтенсивності можна краще розподілити тони на гістограмі, рівномірно використовуючи весь діапазон інтенсивності (рисунок 2.34), що дозволяє ділянкам з низькою локальною контрастністю отримати вищу контрастність.

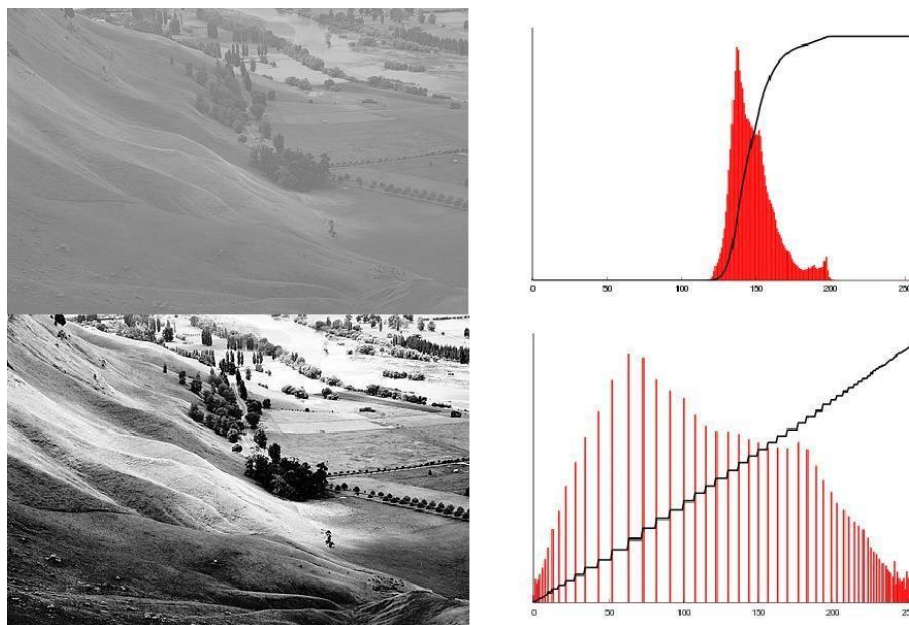


Рисунок 2.34 – Результат методу вирівнювання гістограми

Метод є корисним для зображень, де сцена і об'єкти на передньому плані є яскравими або темними. Недоліком методу є його невибірковість, що

характеризується збільшення контрасту фонового шуму при зменшенні корисної інформативності.

Алгоритм вирівнювання гістограми. Нехай вхідне дискретне зображення представлено одним сірим каналом $\{x\}$, n_i – кількість пікселів рівня інтенсивності n_i . Тоді ймовірність інтенсивності $p_x(i)$ для пікселю i вхідного зображення:

$$p_x(i) = p_x(x = i) = n_i / n, 0 \leq i \leq L \quad (2.44)$$

де

L – кількість можливих рівнів інтенсивності (зазвичай, 256 для 8 бітного представлення),

n – загальна кількість пікселів на зображенні, тоді $p_x(i)$ фактично є гістограмою для інтенсивності пікселя нормований в діапазоні $[0, 1]$.

Функція розподілу (Cumulative Distribution Function, CDF) має вигляд:

$$cdf_x(i) = \sum_j^i p_x(j) \quad (2.45)$$

Введемо функцію перетворення $y = T(x)$, для створення нового зображення y з рівномірною гістограмою. Оскільки CDF зображення лінійне на всьому діапазоні інтенсивності, тоді:

$$cdf_x(i) = (i + 1)K; \quad 0 \leq i < L; \quad (2.46)$$

$$y = T(k) = cdf_x(k); \quad 0 \leq k < L. \quad (2.47)$$

Тому що y нормалізовано в $[0; 1]$, тоді вихідне зображення y масштабуємо до рівня вхідного зображення x :

$$y = \text{ceil}(y(\{x\} - \{x\}) + \{x\}) = \text{ceil}(y(L - 1)) \quad (2.48)$$

При проектуванні архітектурного рішення моделі було обрано алгоритм вирівнювання гістограми для вирішення задачі збільшення контрастності.

2.5 Алгоритм BYTE Track для відстеження багатьох об'єктів

BYTE Track є інноваційним алгоритмом відстеження багатьох об'єктів в реальному часі. Основною перевагою даного методу є велика швидкість роботи відносно інших методів при високій якості відстеження (метрика MOTA, розділ 1.3.4) та ідентифікації об'єктів. Особливо дана перевага стосується відеорядів, де об'єкти частково перекривають один одного, що зумовлено специфікою алгоритму BYTE Track.

Даний алгоритм реалізує концепцію відстеження через розпізнавання, отже для його функціонування необхідний метод для розпізнавання багатьох об'єктів. Найчастіше для задачі розпізнавання використовують натреновані мережі, як реалізують принцип розпізнавання за один прохід (наприклад, YOLO, SSD), що дозволяє максимально реалізувати потенціал швидкодії BYTE Track.

BYTE Track використовує фільтри Калмана для передбачення позиції об'єкту на наступному кадрі відеоряду з урахуванням нормального відхилення. Для кожного відстежуваного об'єкту задається відповідний фільтр Калмана, що передбачає його місцезнаходження на наступному кадрі та оновлюється відповідно до реального місцезнаходження на поточному кадрі після етапу асоціації. Для асоціації об'єктів з попереднього кадру з об'єктами на поточному використовується метрика IoU, що зводить задачу асоціації до типової задачі про призначення. Для вирішення цієї задачі про призначення в стандартному алгоритмі BYTE Track використовувався Угорський метод розв'язання задачі оптимізації.

Угорський метод є алгоритмом комбінаторної оптимізації для розв'язання задачі про призначення, що полягає в пошуку оптимального співставлення між елементами двох множин так, щоб сума ваг цих співставлень була максимальною або мінімальною, залежно від конкретних

умов задачі. Для методу відстеження BYTE Track Угорський алгоритм використовується на етапі асоціації розпізнаних об'єктів на поточному кроці з тими, що були відстеженні на попередніх, або виконує задачу ідентифікації.

2.5.1 Алгоритм методу BYTE Track

Головною ідеєю методу BYTE Track є збереження обмежувальних рамок з малою оцінкою впевненості об'єктів для етапу другої асоціації між поточним та попереднім кадром за оцінкою схожості, що дозволяє не втрачати ці відстежувані об'єкти у відеоряді. Загальний алгоритм дуже адаптивний щодо методу розпізнавання об'єктів, адже приймає на вхід вже результати розпізнавання у вигляді обмежувальних рамок та їх оцінкою впевненості. Інноваційність алгоритму полягає у розділенні об'єктів за рівнем оцінки впевненості на 3 групи: з недостатньою, малою та високою впевненістю та їх окремим опрацюванням.

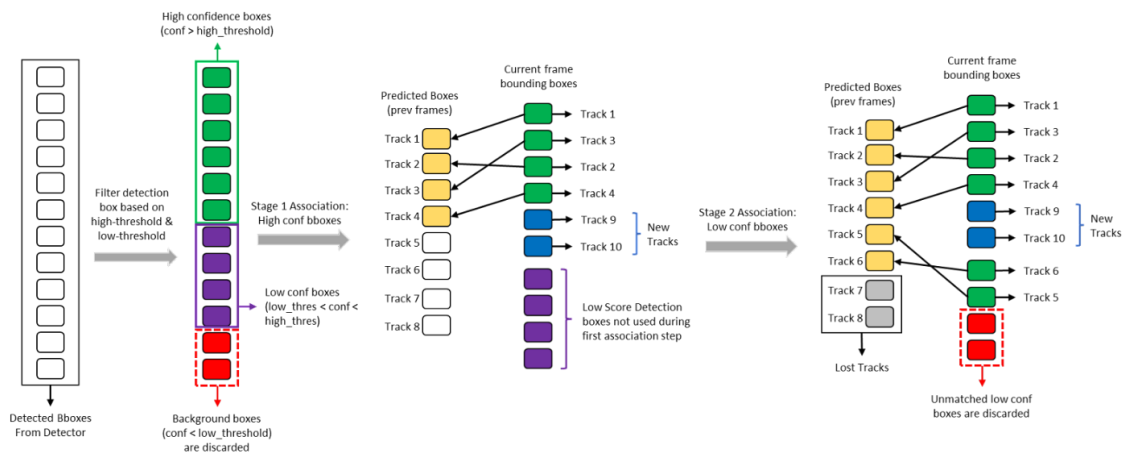


Рисунок 2.35 – Основа алгоритму BYTE Track

Алгоритм методу BYTE Track можна розділити на наступні основні етапи (рисунок 2.35):

1. Розділення вхідних розпізнаних об'єктів $d \in D$ (які представлені обмежувальними рамками) на групи D_{low} і D_{high} : за оцінкою впевненості, використовуючи задані параметри методу $threshold_{low}$ та $threshold_{high}$:

$$\begin{aligned} d &\in D_{high}, \text{ якщо } conf \in [threshold_{high}, 1] \\ d &\in 1, \text{ якщо } conf \in [threshold_{low}, threshold_{high}) \\ d &\text{ – відкидаються, якщо } conf \in [0, threshold_{low}) \end{aligned} \quad (2.49)$$

2. Передбачення для всіх об'єктів, які відстежувалися ($t \in T$) на попередніх кадрах, їх положення на поточному кадрі, використовуючи фільтри Калмана.

3. Асоціація розпізнаних об'єктів ($d \in D_{high}$) з високою оцінкою впевненості з попередньо відстежуваними об'єктами. Асоціація відбувається через Угорський алгоритм, використовуючи метрику IoU між розпізнаними об'єктами D_{high} і результатом передбачення фільтрів Калмана (у вигляді обмежувальної рамки). Об'єкти з D_{high} , яким не знайшлося асоціації, вважаються новими та поповнюють множину T (в кінці ітерації).

4. Асоціація розпізнаних об'єктів ($d \in D_{low}$) з високою оцінкою впевненості з відстежуваними об'єктами, яким не знайшлося асоціації на попередньому кроці. Розпізнані об'єкти, яким не знайшлося асоціації відкидаються як малоймовірні, а відстежуванні об'єкти ($t \in T$) без асоціації – є втраченими.

5. Оновлення фільтрів Калмана.

2.5.2 Фільтр Калмана

Фільтр Калмана (Kalman filter), або лінійно-квадратичне оцінювання (linear quadratic estimation, LQE), представляє собою алгоритм, який обробляє послідовності вимірювань, отриманих протягом часу, із врахуванням шумів та інших випадкових відхилень. Алгоритм генерує оцінки невідомих змінних, які

можуть бути потенційно точнішими, ніж ті, що базуються виключно на вимірюваннях. З формальної точки зору, фільтр Калмана рекурсивно працює з зашумленими вхідними даними та видає статистично оптимальні оцінки базового стану системи.

Цей алгоритм працює як двоетапний процес:

- на етапі передбачення надає оцінки змінних поточного стану разом із їхніми невизначеностями;
- при отриманні спостереження від наступного вимірювання, яке завжди містить певне відхилення, включаючи випадковий шум, ці оцінки коригуються за допомогою середнього зваженого, де більша вага надається більш точним оцінкам.

Рекурсивний характер алгоритму дозволяє йому працювати в режимі реального часу, використовуючи лише наявні вхідні дані, попередньо розрахований стан та матрицю невизначеності, без необхідності додаткової інформації.

Базова модель. Фільтри Калмана, ґрунтуючись на дискретизованих у часі лінійних динамічних системах, використовують моделі у вигляді ланцюгів Маркова, що побудовані на лінійних операторах із забрудненням помилками, що можуть включати гаусів шум (рисунок 2.36). Стан системи виражається вектором дійсних чисел і на кожному такті дискретного часу до стану застосовується лінійний оператор для генерації нового стану із додаванням деякого шуму та інформації від систем управління (при наявності). Надалі інший лінійний оператор, який комбінований з додатковим шумом, застосовується до справжнього (прихованого) стану для генерації спостережуваних виходів.

Для використання фільтра Калмана для оцінки внутрішнього стану процесу, виходячи лише із послідовності зашумлених спостережень, необхідно відповідно змодельовати процес за моделлю динамічної системи для фільтра Калмана. Це включає задання матриць F_k (модель переходу станів),

H_k (модель спостереження), Q_k (коваріація шуму процесу), R_k (коваріація шуму спостереження) та за потреби B_k (модель управління) для кожного моменту часу k .

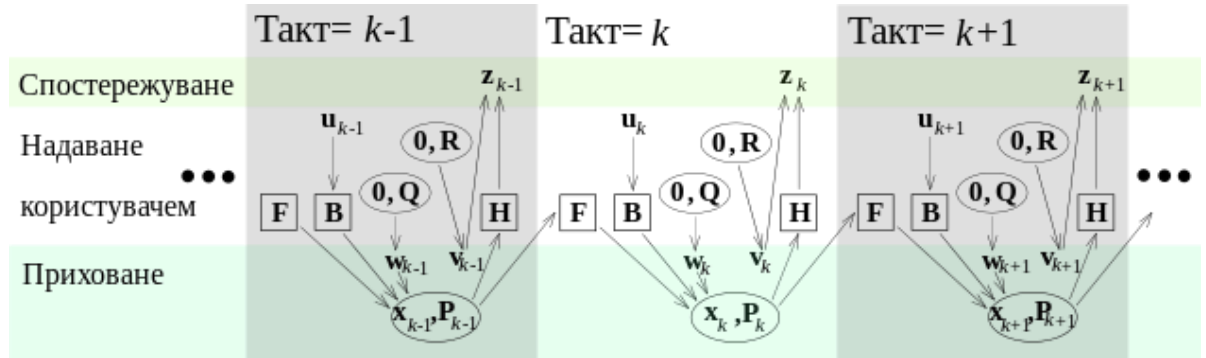


Рисунок 2.36 – Базова модель динамічної системи для фільтра Калмана

Модель фільтра Калмана припускає, що справжній стан у момент часу k виводиться зі стану в $(k - 1)$:

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (2.50)$$

де

F_k – є моделлю переходу стану, що застосовується до попереднього стану x_{k-1} ;

B_k – є моделлю впливу керування на систему, що застосовується до вектора керування u_k ;

$W_k \sim N(0, Q_k)$ – є шумом процесу, що, як вважається, має багатовимірний нормальний розподіл з нульовим середнім значенням і з коваріацією Q_k .

У момент часу k спостереження (або вимірювання) z_k справжнього стану x_k робиться відповідно:

$$z_k = H_k x_k + v_k \quad (2.51)$$

де

H_k – модель спостереження, що відображає простір справжнього стану у спостережуваний простір;

v_k – шум спостереження, що, як вважається, є гаусовим білим шумом з нульовим середнім значенням і з коваріацією R_k .

При цьому початковий стан і вектори шуму на кожному такті вважаються взаємно незалежними.

Загальний алгоритм. Фільтр Калмана є рекурсивним оцінювачем, який прийнято ділити на 2 етапи: передбачення стану системи, використовуючи стан системи з попереднього такту та уточнення, який використовує результати спостереження на поточному такті. Стан системи представлений двома змінними: $\hat{x}_{k|k}$ – апостеріорна оцінка стану системи в момент часу k при відомих спостереженнях до моменту k включно; $P_{k|k}$ – апостеріорна коваріаційна матриця помилок, що є мірою оцінки точності отриманої оцінки стану.

Етап передбачення представляє вирахування апіорної оцінки системи $\hat{x}_{k|k-1}$ та коваріацію апіорної оцінки $P_{k|k-1}$:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \quad (2.52)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (2.53)$$

Етап уточнення передбачає об'єднання апіорної оцінки після етапу передбачення з поточною інформацією спостереження:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \quad (2.54)$$

$$S_k = H_k P_{k|k-1} H_k^t + R_k \quad (2.55)$$

$$K_k = P_{k|k-1} H_k^t S_k^{-1} \quad (2.56)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} - K_k \tilde{y}_k \quad (2.57)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (2.58)$$

де

\tilde{y}_k – нововведення (відхилення) вимірювання,
 S_k – коваріація нововведень (відхилення),
 K_k – оптимальний передавальний коефіцієнт Калмана,
 $\hat{x}_{k|k}$ – оновлена (апостеріорна) оцінка стану,
 $P_{k|k}$ – коваріація оновленої (апостеріорної) оцінки,
 I – одинична матриця.

Застосування при задачі відстеження. Застосування фільтрів Калмана для вирішення задачі відстеження об'єктів вимагає специфікувати загальний алгоритм. Для спрощення опису алгоритму буде розглянуто спрощену задачу відстеження в одновимірному просторі (рис 2.37). Нехай, спостереження руху об'єкту вимірюється з періодом в Δt секунд й ці виміри є неточними (у випадку відстеження багатьох зображень за відеорядом, причиною неточності є помилка в визначенні обмежувальної рамки об'єкту, й відповідно координати його центру). Так як значення F , H , R , Q – сталі, то їх індекси в подальшому описі будуть опущені.

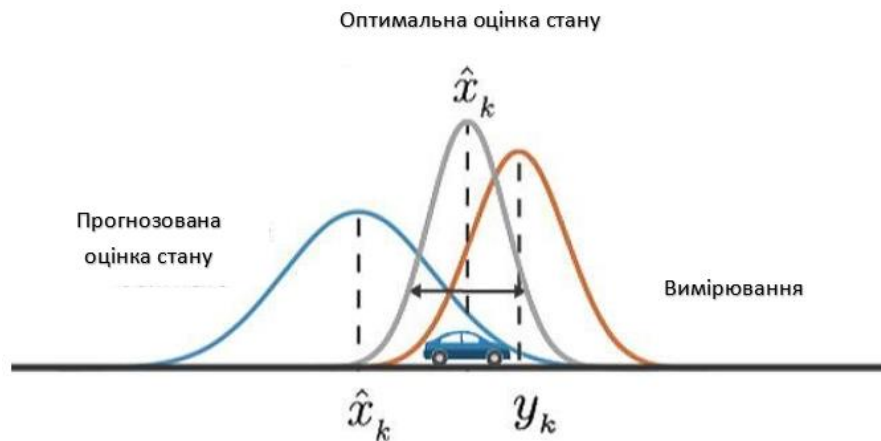


Рисунок 2.37 – Застосування фільтру Калмана в задачі відстеження

Таким чином, положення та швидкість об'єкту вимірюється лінійним простором стану:

$$x_k = [x \dot{x}] \quad (2.59)$$

де \dot{x} – швидкість (похідна положення по часу).

Нехай зміну швидкості об'єкта описує прискорення a_k , що має нормальний розподіл із нульовим середнім значенням, та стандартним відхиленням $a_k \sim N(0, \sigma_a)$. Тоді передбачення стану системи представлятиме у векторному виді кінетичного рівняння:

$$x_k = Fx_{k-1} + Ga_k. \quad (2.60)$$

При:

$$F = [1 \ \Delta t \ 0 \ 1] \quad (2.61)$$

$$G = \left[\frac{\Delta t^2}{2} \ \Delta t \right] \quad (2.62)$$

З загального алгоритму фільтру Калмана:

$$x_k = Fx_{k-1} + w_k \quad (2.63)$$

$$w_k \sim N(0, Q) \quad (2.64)$$

$$Q = GG^T \sigma_a^2 = \left[\frac{\Delta t^4}{4} \ \frac{\Delta t^3}{2} \ \frac{\Delta t^3}{2} \ \Delta t^2 \right] \sigma_a^2 \quad (2.65)$$

На кожному кроці здійснюється зашумлене вимірювання z_k справжнього положення об'єкту. Припускаючи, що шум вимірювання v_k також має нормальний розподіл із нульовим математичним очікуванням та стандартним відхиленням σ_z , то:

$$z_k = H_k x_k + v_k \quad (2.66)$$

$$H = [1 \ 0]^T \quad (2.67)$$

$$R = E[v_k v_k^T] = \sigma_z^2 \quad (2.68)$$

Етап передбачення та оновлення співпадає з загальним алгоритмом фільтру Калмана, описаним вище.

При розробленні власної моделі використано готове програмне рішення з реалізацією фільтра Калмана та Угорського алгоритму.

2.6 Висновки до розділу 2

В даному розділі проведено ґрунтовне дослідження окремих аспектів та алгоритмів вибраних методів препроцесінгу, виявлення, розпізнавання та відстеження об'єктів.

Зокрема, розглянуто алгоритми сімейства YOLO та Faster R-CNN, для яких описано деталі їх навчання та наведено результати досліджень.

Основну увагу приділено питанню еволюції алгоритму YOLO, що є ключовим для розуміння структури даного алгоритму та використаних методів та підходів організації даних.

Детально розглянуто алгоритм розпізнавання YOLOv8, як поточний еталон для моделей розпізнавання.

Додатково розглянуто алгоритм розпізнавання YOLOv9 з його комплексною функцією втрат та топологією загальної мережі з описом основних блоків, який є цілком достатнім для його подальшої реалізації, що дозволяє практично побудувати його екземпляр через розкрите архітектурне рішення. Опис складових функцій втрат з її параметрами надає розуміння про оцінку роботи мережі під час її навчання. Алгоритм дозволяє правильно інтерпретувати вихідні дані з мережі та вирішувати базові проблеми розпізнавання об'єктів на зображенні.

Наведено наявні проблеми поганої якості зображення та показано можливість їх вирішення (препроцесинг зображень) запропонованими методами.

Розроблено архітектуру моделі YOLO v9 P (на базі моделі YOLOv9), яка заснована на модульному підході, демонструє адаптивність до змінних умов поведінки об'єктів в режимі реального часу, що забезпечує стабільну роботу програмного засобу.

По більшості параметрів запропоновано та побудовано метод виявлення, розпізнавання та відстеження об'єкту в режимі реального часу YOLO v9 P,

який видає більш точні результати при затраті меншої кількості ресурсів у порівнянні з YOLO v9, що дозволяє використовувати в системах з обмеженими ресурсами.

Теоретично обґрунтовано такі причини поганої якості зображень, як шум, розмиття та недостатня контрастність та розкрито методи для вирішення цих проблем: медіанний фільтр та TV-регуляризація для видалення шуму, фільтр чіткості та Deblur GAN для видалення розмиття та вирівнювання гістограм для підвищення контрастності.

Для подальшого конструювання програмного рішення обрано метод BYTE Track для відстеження багатьох об'єктів в реальному часі, що складається з передбачення положення об'єктів через фільтри Калмана та розділення ідентифікованих об'єктів за рівнем впевненості, первинної та вторинної асоціації.

Основними змінами запропонованої архітектури моделі YOLO v9 P є додаткові блоки Detect модулю Head, які відповідають за вирішення задачі виявлення об'єктів малого розміру; додаткові блоки CBLiner, CBFuse, ADown та RepNCSPELAN4 модулю Auxiliary для агрегації шарів в структуру ELAN і спрямування на подальшу оптимізацію процесу виділення ознак; додаткові блоки PFNB і BiFormer модулю Backbone для зменшення обчислювальної складності та покращення уваги моделі до ключової інформації у вхідних ознаках.

РОЗДІЛ 3. РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСОБУ РОЗПІЗНАВАННЯ ТА ВІДСТЕЖЕННЯ

У даному розділі описано процес підготовки набору даних (dataset), обґрунтування середовища та технологій розроблення програмного продукту, готові програмні модулі та результати експерименту. Для навчання нейронної мережі розпізнавання необхідно обрати набір даних, що відповідає критеріям, які сформовані до завдання дисертаційного дослідження та має якісну перевагу над іншими існуючими роботами.

Для реалізації програмного рішення та проведення експерименту необхідно обрати технології, які дозволяють зручно реалізувати описані методи та середовище, що є достатньо потужні для проведення якісного експерименту. Реалізована система ділиться на модулі препроцесингу, розпізнавання та відстеження, що містять відповідний функціонал.

Експеримент представляє собою проведення замірів визначених метрик та часу виконання для системи з різними методами препроцесингу.

3.1 Огляд існуючих наборів даних (dataset)

Відповідно до поставленого завдання набір даних для реалізації та валідації системи розпізнавання та відстеження повинен задовольняти наступним критеріям:

- джерелом зображення повинен бути БпЛА з висотою польоту на момент зйомки до 50 м, бажано роторного типу;
- на зображеннях повинні бути об'єкти техніки різних класів (при можливості, люди також мають бути) з відповідною розміткою для кожного об'єкта зображення;
- зображення з розміткою мають складати разом послідовність кадрів відеоряду з частотою зміни кадрів (FPS, frames per second) ≥ 10 .

Набір даних Kitty з розділу 2.1 не відповідає переліченим критеріям для реалізації та валідації системи розпізнавання об'єктів з БпЛА, але планується використати як еталонний рекомендований набір даних в моделі при перевірці алгоритмів виявлення об'єктів.

Порівняльну характеристику наборів даних з БпЛА наведено у таблицях 3.1 та 3.2.

Таблиця 3.1 – Порівняльна таблиця наборів даних з БпЛА

Назва набору даних	Джерело зображень	Рік
COWC	Aerial	2016
CARPK	Drone	2017
DOTA	aerial	2018
UAVDT	Drone	2018
VisDrone	Drone	2018
DroneVehicle	Drone	2022
HIT-UAV	Drone	2022
RAIVD	Drone	2022
TAIVD	Drone	2022

Якщо для перших двох критеріїв існує достатньо датасетів у вільному доступі (таблиці 3.1 – 3.2), то під третій критерій підлягає лише два з них, а саме: TAIVD «Traffic Aerial Images for Vehicle Detection» і RAIVD «Roundabout Aerial Images for Vehicle Detection».

Обидва набори даних мають приблизно однакову кількість кадрів та однакове розширення та тип зображення. RAIVD містить кадри відеоряду з дрона, зняті на більшій висоті, ніж TAIVD, що приводить до більшої частоти об'єктів на одному зображенні, але також і призводить до проблеми малого масштабу об'єктів. Також у TAIVD є відеоряди із зміною плану зйомки та

кутом відеокамери, коли в RAIVD. Кількість унікальних локацій також більша в TAIVD, ніж у RAIVD (12 та 8 відповідно).

Таблиця 3.2 – Порівняльна таблиця наборів даних з БпЛА

Назва набору даних	Тип зображень	Кількість зображень	Кількість класів	Середня кількість об'єктів одного класу	Розширення
COWC	RGB	32.7k	1	32.7k	2048x2048
CARPK	RGB	1.448	1	89.8k	1280x720
DOTA	RGB	2.806	14	13.4k	12029x5014
UAVDT	RGB	80k	3	280.5k	1080x540
VisDrone	RGB	10,209	10	54.2k	2000x1500
DroneVehi cle	RGB+ Infrared	31,064	5	88.3k	840x712
HIT-UAV	HIT	2,898	4	5.3k	640x512
RAIVD	RGB	15,474	5	62.5k	1920x1080
TAIVD	RGB	15,070	2	75.2k	1920x1080

Об'єднати дані набори даних неможливо через часткове неспівпадіння в класифікації об'єктів. В датесеті RAIVD є клас «велосипед», а в TAIVD – немає, але є клас «мотоцикл», якого немає в першому, також є різниця в розмітці класу «автомобіль». Отже, незважаючи на більшу кількість класів в RAIVD, для подальшого дослідження було обрано набір даних TAIVD. Однією з основних причин такого вибору є більша різноманітність планів відеозйомки.

Опис набору даних TAIVD. Дані (рисунок 3.1) були зібрані в 2022 році у вигляді відеорядів з 2 типів комерційних БпЛА (DJI Mavic Mini 2 та Yuneec

Турпоон Н) для польотів на малі та середні відстані [47]. Об'єктами зйомки стали ділянки дороги, перехрестя та перехрестя з круговим рухом разом з транспортом на них. Локація зйомки – Іспанія, більшість відеорядів була знята на околицях Мадриду. Кут напрямку камери до горизонту коливається від 45° до 60°. Цей діапазон дозволяє захоплювати бокові та верхні частини об'єктів, на відміну від повністю зенітного захоплення під кутом 90°, коли буде взята лише верхня проекція.

Висота польоту БпЛА при зборі даних коливається з 35 м до 120 м. Що дозволило отримати знімки різних масштабів і збільшило різноманіття набору даних.

Отриманні відеоряди мали частоту кадрів 30 FPS та розширення в 1920×1080 пікселів. Зображення збережені у файлі формату .jpg, назва якого складається з назви відеоряду та порядкового номеру кадру у відеоряді.

Розмітка даних для об'єктів було виконано мануально авторами TAIVD у форматі YOLO, через його уніфікованість та високу поширеність для задачі розпізнавання об'єктів. Для кожного кадру було сформовано текстовий файл з назвою відповідно до назви зображення кадру. Кожен рядок файлу відповідає одному об'єкту на зображенні й містить інформацію про його клас та обмежувальну рамку у форматі, що наведено у формулі 3.1.

$$< id_{class} > < x > < y > < w > < h > \quad (3.1)$$

де

id_{class} – ідентифікатор класу об'єкту (0 – для класу «автомобіль», 1 – для «мотоцикл»),

x, y – значення координатів центру обмежувальної рамки, нормалізована у діапазоні [0.0, 1.0],

w, h – значення ширини і висоти обмежувальної рамки відносно ширини й висоти зображення (також нормалізований в [0.0, 1.0]).



Рисунок 3.1 – Приклад кадру відеоряду з зображеними обмежувальними рамками, колір яких відповідає за клас об'єкту

3.2 Формування власного датасету

З огляду на те, що практичним аспектом роботи є виявлення в режимі реального часу саме військової техніки, а датасетів з таким маркуванням у відкритих джерелах не виявлено, було прийняте рішення про формування власного датасету. Для цього було використано фото та відео матеріали із відкритих джерел, а саме відеороликів на відео хостинговій платформі Youtube, соц. Мережі facebook, та в месенджерах Telegram, Viber та WhatsApp. Основною проблемою стала погана якість великої кількості роликів, наявність шумів. Також часто в таких відео фігурують ватермарки, які часто перекривають області інтересу, цим самим знижуючи якість потенціального датасету. Проте підбір матеріалів це лише перша частина створення датасету, іншою ж частиною є розмітка та маркування.

У світі штучного інтелекту та машинного навчання, що постійно розвивається, попит на високоякісні марковані дані продовжує зростати в геометричній прогресії. Будь то навчання моделей комп'ютерного зору для розпізнавання об'єктів на зображеннях або тонка настройка алгоритмів

опрацювання даних природної мови для аналізу настроїв, марковані набори даних необхідні для розроблення та оптимізації програмного забезпечення додатків штучного інтелекту. На цьому тлі платформи для анотування даних, такі як Label Studio і Roboflow, стали популярним вибором для оптимізації процесу маркування даних.

Надалі проведено порівняльний аналіз Label Studio і Roboflow, висвітлено їхні особливості, функціональні можливості та придатність для різних сценаріїв використання, щоб допомогти прийняти обґрунтоване рішення при виборі правильного інструменту для потреб в анотуванні даних.

3.2.1 Вибір програмного інструментарію для формування набору даних

Roboflow є хмарною платформою, що спеціалізується на завданнях комп'ютерного зору, таких як виявлення об'єктів, класифікація зображень і сегментація зображень. Завдяки зручному інтерфейсу та безшовній інтеграції з популярними фреймворками машинного навчання, такими як TensorFlow та PyTorch, Roboflow спрощує процес анотування та попередньої обробки наборів даних зображень для навчання моделей комп'ютерного зору. Roboflow пропонує ряд інструментів для анотування, включаючи обмежувальні рамки, полігони та ключові точки, а також автоматизовані методи доповнення даних для збільшення різноманітності наборів даних і підвищення надійності моделей.

Функції та можливості Roboflow (рисунок 3.2) надають розробникам корисних інструментів для створення анотацій, автоматизацією попередньої обробки та доповнення даних, безшовну інтеграцію з популярними фреймворками та платформами, співпрацю та контроль версій.

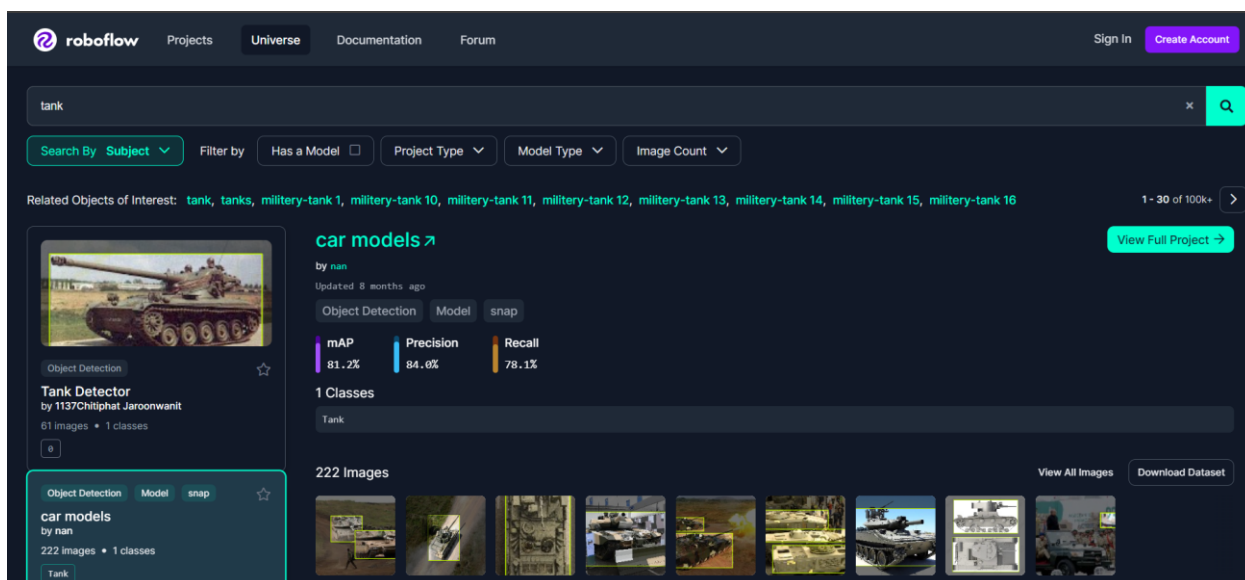


Рисунок 3.2 – Приклад набору даних Roboflow

Label Studio є інструментом для маркування даних з відкритим вихідним кодом, розроблений Heartex. Завдяки підтримці широкого спектру типів даних, включаючи текст, зображення, аудіо та відео, Label Studio пропонує уніфіковану платформу для анотування різноманітних наборів даних у різних галузях. Однією з особливостей Label Studio є гнучкість у створенні власних інтерфейсів анотацій, пристосованих до конкретних сценаріїв використання та вимог до анотацій. Завдяки зручному інтерфейсу та налаштовуваним конфігураціям маркування, Label Studio дає можливість як фахівцям з даних, так і нетехнічним користувачам ефективно маркувати дані для різних завдань машинного навчання.

Ключові особливості Label Studio є мультимодальну підтримку, гнучкі анотації, інструменти для співпраці, інтеграцію з активним навчанням, масштабованість. Під час додавання анотацій Label Studio автоматично впорядковує марковані дані, що полегшує відстеження прогресу та керування проєктами анотацій (рисунок 3.3). Платформа також підтримує роботу з версіями, дозволяючи користувачам переглядати і порівнювати анотації в різних ітераціях.

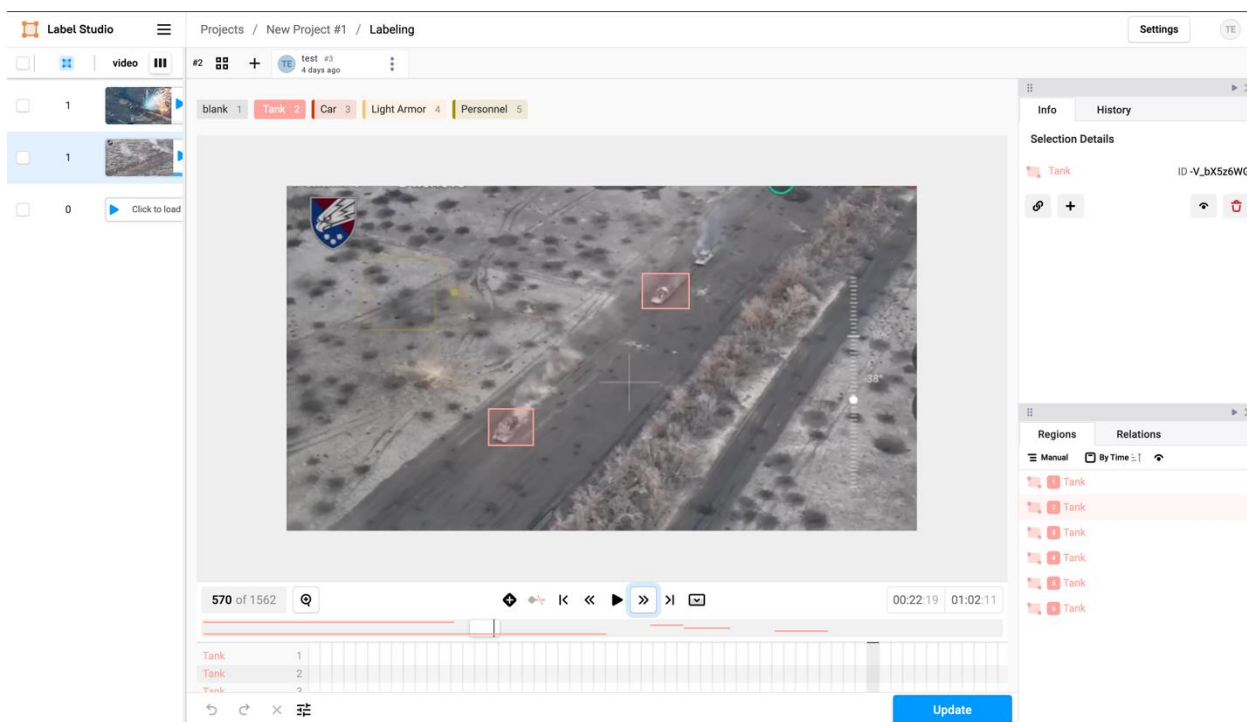


Рисунок 3.3 – Приклад кадру відеоряду в Label Studio

Переваги використання Label Studio є підвищення ефективності, покращена якість, економічна ефективність, гнучкість і кастомізація, підтримка спільноти, спеціальні інтерфейси анотацій, співпраця та командна робота, інтеграція з активним навчанням, масштабованість і продуктивність.

3.2.2 Порівняльний аналіз Label Studio та Roboflow для анотування даних

Гнучкість проти спеціалізації: Label Studio пропонує гнучкість у роботі з різними типами даних і завданнями анотацій, що робить його придатним для широкого спектру застосувань III. Roboflow, навпаки, спеціалізується на завданнях комп'ютерного зору і може бути більш обмеженим у застосуванні в інших сферах.

Кастомізація проти автоматизації: Label Studio надає пріоритет кастомізації, дозволяючи користувачам створювати індивідуальні інтерфейси

анотацій і робочі процеси. Roboflow, з іншого боку, робить акцент на автоматизації з такими функціями, як автоматичне доповнення даних і конвеєри попередньої обробки, призначені для впорядкування робочих процесів комп'ютерного зору.

Відкритий код проти хмарних технологій: Label Studio має відкритий вихідний код і може бути розміщений на власному хостингу, пропонуючи більший контроль і гнучкість над розгортанням та інфраструктурою. Roboflow – хмарна платформа, що пропонує зручність і масштабованість, але потенційно обмежує можливості кастомізації для користувачів з особливими вимогами до розгортання.

Співпраця та інтеграція: і Label Studio, і Roboflow підтримують співпрацю між членами команди з функціями призначення завдань, відстеження прогресу та перевірки анотацій. Крім того, обидві платформи пропонують інтеграцію з популярними фреймворками машинного навчання, що полегшує безперебійне навчання моделей і робочі процеси розгортання.

Таким чином, Label Studio і Roboflow – це потужні інструменти для анотування даних, кожен з яких пропонує унікальні функції та сильні сторони, пристосовані до різних сценаріїв використання та вподобань. У той час як Label Studio вирізняється гнучкістю та кастомізацією, задовольняючи широкий спектр типів даних і завдань анотування, Roboflow спеціалізується на впорядкуванні робочих процесів комп'ютерного зору за допомогою автоматизованих інструментів і хмарної інфраструктури. Зрештою, вибір між Label Studio і Roboflow залежить від конкретних вимог, цілей проєкту та уподобань щодо кастомізації, автоматизації та розгортання. Ретельно оцінивши можливості та функції кожної платформи, було прийняте рішення використовувати Label Studio через можливість розгорнути його локально, таким чином створений датасет не зможе потрапити у відкритий доступ через проблеми та вразливості хмарного середовища, або через особливості політики його використання.

3.3 Обґрунтування технологій та середовища для розроблення програмного забезпечення

Вибір технологій та середовища розроблення програмного забезпечення в значній мірі залежить від поставленої задачі. На даний вибір також впливають функціональні та нефункціональні вимоги до системи програмного продукту. У сфері машинного навчання загалом та для задач розпізнавання та відстеження об'єктів за відеорядом зокрема найбільш популярними є наступні мови програмування.

Julia є відносно новою високорівневою мовою програмування, яку найчастіше використовують для аналізу числових даних. Основною перевагою Julia є висока швидкість виконання (майже як на мові C), основним недоліком є недостатня спільнота розробників, що для мов з відкритим кодом є важливим фактором.

R є однією з популярних мов для задач машинного навчання, яка спеціалізована для статистичних досліджень. Має простий синтаксис, але невелику сферу використання через доволі вузьку спеціалізацію.

Наразі найбільш зручною мовою програмування для створення моделей нейронних мереж є Python через простий інтуїтивно-зрозумілий синтаксис, простоту використання та широке поширення серед розробників, яку і було використано для виконання практичної частини даної роботи. Для Python розроблено багато фреймворків та бібліотек з відкритим кодом для різних задач по роботі з даними, візуалізації, машинного навчання та розробки нейронних мереж, а велика спільнота розробників забезпечують стабільність роботи та актуальність.

Ірму для оптимізації написання власного програмного забезпечення використано ряд бібліотек та фреймворків, основні з яких представлено нижче:

– Pytorch є фреймворком для різних задач машинного навчання на основі бібліотеки Torch, який став популярний для задач проектування глибоких нейронних мереж; перевагою Pytorch є реалізація модулів для створення й навчання нейронних мереж, оптимізованих на мові C та підтримка інтерфейсу CUDA для використання графічних процесорів GPU;

– Ultralytics є модулем для створення, навчання та розгортання для моделей комп'ютерного зору, написаний на основі Pytorch; даний модуль відомий зручною реалізацією ряду мереж архітектури YOLO;

– Numpy є модулем для роботи з багатовимірними масивами (найчастіше використовувався для матриць та тензорів);

– OpenCV2 є бібліотекою для операцій з зображенням, їх читання та збереження, робота з відеорядом, функціонал якої оптимізований на мові програмування C;

– Matplotlib є бібліотекою, яка використовувалась для створення та візуалізації графіків.

Вирішення задач комп'ютерного зору потребують достатньо великих обчислювальних ресурсів машини, на якій буде розгорнуто середовище. Навчання комплексних нейронних мереж на датесеті із зображень вимагатиме проведення великої кількості обчислень. Параметри ноутбука для навчання нейронної мережі наведено в таблиці 3.3.

Таблиця 3.3 – Опис параметрів ноутбука для навчання нейронної мережі

Параметри	Конфігурація
CPU	Intel Core i7-9750H
RAM	32 GB DDR4 2400 MHz
GPU	Intel UHD Graphics 630 4 Gb + AMD Radeon Pro 560 X 4 Gb GDDR5
Операційна система (ОС)	Mac OS

3.4 Результати експериментів

3.4.1. Порівняльні експерименти з функціями втрат різного вигляду

Для демонстрації ефекту збільшення ефективності роботи покращеної моделі проведено імітація моделювання порівняльних експериментів між покращеною та базовою моделями YOLO v9 при невеликому наборі даних (на невеликій кількості зображень) з урахуванням доступних обчислювальних потужностей дослідника з можливістю подальшого масштабування.

Щоб перевірити перевагу впровадження WIoU v3, проведено моделювання порівняльних експериментів при невеликому наборі даних на YOLO v9 з використанням WIoU v3 та деяких основних функцій втрат, зберігаючи інші умови навчання незмінними.

Результати експериментів наведено в таблиці 3.4.

Таблиця 3.4 – Порівняння результатів виявлення для різних функцій втрат

Метрика	Точність (P), %	Повнота (R), %	mAP 0.5, %	mAP 0.5:0.95, %
CIoU	50.9	38.2	39.3	23.5
DIoU	51.0	38.3	39.5	23.6
GIoU	50.3	38.4	39.6	23.6
EIoU	49.1	38.0	38.7	23.4
SIoU	51.5	38.5	39.4	23.4
WIoU v1	50.1	38.5	39.3	23.3
WIoU v2	50.6	38.4	39.3	23.2
WIoU v3	51.3	38.6	40.0	23.6

Експериментальні результати показують, що модель досягає найкращої ефективності виявлення при використанні WIoU v3 як функції регресійних

втрат. Крім того, значення mAP моделі при використанні WIoU v3 на 0,7% вище, ніж при використанні CIoU, що свідчить про ефективність використання WIoU v3.

У таблиці 3.4 наведено значення AP для кожної категорії та значення mAP0.5 для всіх категорій для покращеної моделі YOLO v9 P та звичайним YOLO v9.

3.4.2. Порівняння результатів модифікованого алгоритму YOLO v9 P з базовим YOLO v9

Як видно з результатів порівняння в таблиці 3.5, значення mAP для покращеної моделі на невеликому наборі даних покращено на 7,7%. Значення AP для всіх категорій покращуються різною мірою, а значення AP для трьох категорій (танк, люди та транспорт) покращуються більш ніж на 10%. Це свідчить про те, що покращена модель може ефективно підвищити точність виявлення малих об'єктів та покращити продуктивність виявлення.

Таблиця 3.5 – Порівняння запропонованої вдосконаленої моделі та точності виявлення YOLO v9 P

Модель	Пішохід	Люди	Велосипед	Машина	Фура	Вантажівка	Автобус	mAP
YOLO v9	42.7	32.0	12.4	79.1	44.0	36.5	57.0	39.3
YOLO v9 P	56.8	44.9	18.8	85.8	50.8	39.0	64.3	47.0

На рисунках 3.4, 3.5 та 3.6 показано криві зміни деяких важливих оціночних метрик запропонованої нами моделі YOLO v9 P та YOLO v9 у процесі навчання. Тренувальна крива YOLO v9 P та YOLO v9 оціночної

метрики усередненої точності mAP наведено на рисунку 3.4. Тренувальна крива YOLO v9 P та YOLO v9 оціночної метрики точності P наведена на рисунку 3.5. Тренувальна крива YOLO v9 P та YOLO v9 оціночної метрики повноти R наведена на рисунку 3.6.

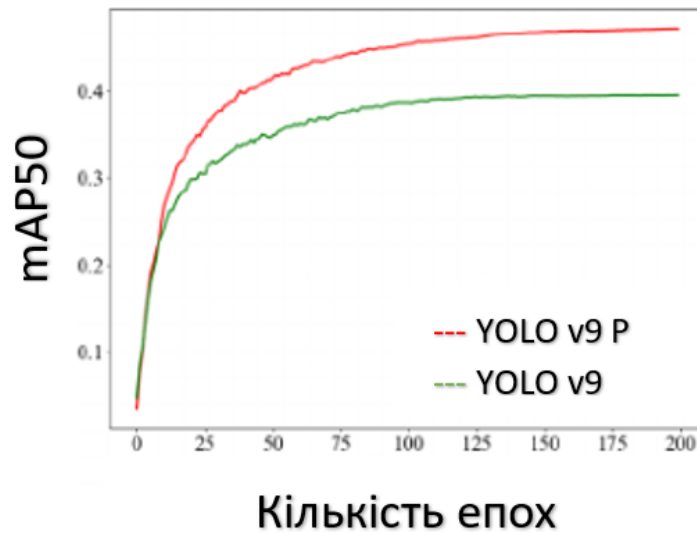


Рисунок 3.4 – Тренувальна крива YOLO v9 P та YOLO v9 оціночної метрики усередненої похибки (mAP)

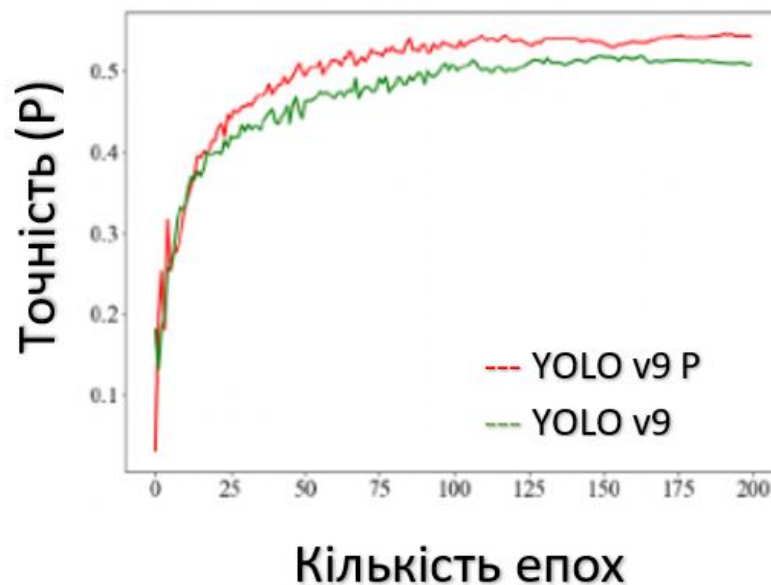


Рисунок 3.5 – Тренувальна крива YOLO v9 P та YOLO v9 оціночної метрики точності (P)

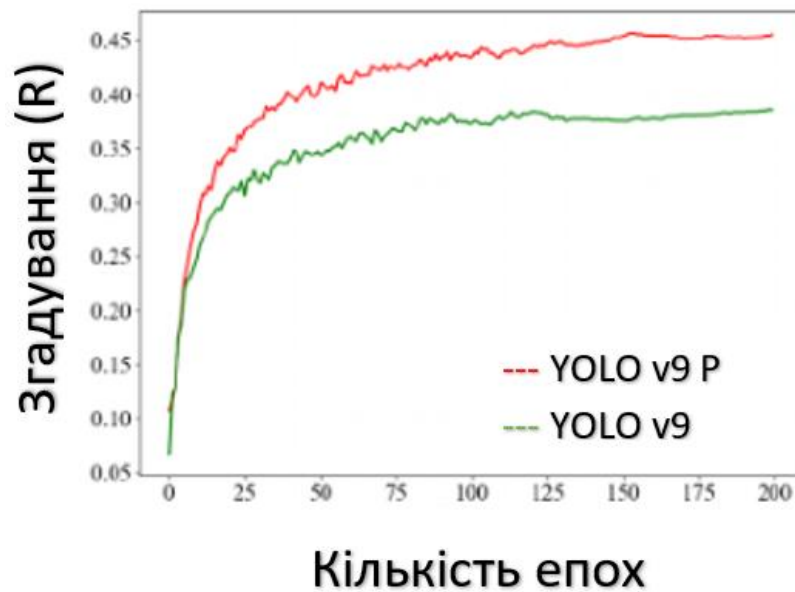


Рисунок 3.6 – Тренувальна крива YOLO v9 P та YOLO v9 оціночної метрики повноти (R)

З рисунків 3.4, 3.5 та 3.6 видно, що запропонована модель YOLO v9 P перевершує YOLO v9 за трьома метриками виявлення: точністю, повнотою та $mAP_{0.5}$ приблизно через 15 епох навчання від початку, а розроблена модель починає стабілізуватися приблизно через 200 епох навчання. Порівняно з YOLO v9, запропонований метод швидше навчається і краще виявляє. Для більш точних результатів повинно провести експеримент в 500 епох.

Для подальшої демонстрації ефективності запропонованого методу порівняно запроповану модель з кількома різними моделями YOLO v8 (YOLO v8n, YOLO v8s, YOLO v8m), YOLO v9) на наборі даних TAIVD. Результати експерименту наведено в таблиці 3.6.

Згідно з даними таблиці 3.6, порівняно з іншими моделями, покращена модель YOLO v9 P має найвищі значення трьох оціночних індексів повнота (R), $mAP_{0.5}$ та $mAP_{0.5:0.95}$, а ефективність виявлення є кращою, ніж у моделі з більшим розміром, ніж у неї самої.

Таблиця 3.6 – Порівняння розробленої моделі YOLO v9 P з існуючими YOLO v9, YOLO v8n, YOLO v8s, YOLO v8m

Модель	Точність (P), %	Повнота (R), %	mAP0.5, %	mAP0.5:0.95, %	Розмір моделі, мс	Час виявлення, мс	Кількість параметрів, 10^6
YOLO v8n	43.8	33.0	33.3	19.3	6.6	4.2	3.0
YOLO v8s	50.9	38.2	29.3	23.5	22.5	7.7	11.1
YOLO v8m	56.0	42.5	44.6	27.1	49.6	16.6	25.9
YOLO v9	57.5	44.3	46.5	28.7	83.5	25.6	43.7
YOLO v9 P	54.4	45.6	47.0	29.2	21.5	19.5	10.3

З результатів експерименту видно, що запропонована п'ятимасштабна структура виявлення може покращити точність виявлення малих об'єктів. Крім того, запроваджений нами механізм уваги BiFormer, що потребує малої обчислювальної потужності, покращує ефективність виявлення моделі, не споживаючи при цьому надто багато ресурсів.

3.4.3. Додавання блоку BiFormer

Для отримання найкращої продуктивності та полегшення подальших експериментів з абеляції після додавання блоку BiFormer до моделі, в цьому дослідженні були проведені наступні порівняльні експерименти. Використовували YOLO v9 P після впровадження WIoU v3 як базову модель і замінили модуль C2f на блок BiFormer на різних рівнях магістральної мережі, і отримали експериментальні результати, що наведені в таблиці 3.7.

Знак "+" у таблиці 3.7 означає, що блок BiFormer додано до базової моделі. B3-BiFormer вказує на те, що модуль C2f між шаром B3 і шаром B4

замінено блоком BiFormer, а B4-BiFormer вказує на те, що модуль C2f між шаром B4 і шаром B5 замінено блоком BiFormer.

Згідно з результатами експерименту на невеликому наборі даних, наведеними в таблиці 3.7, найкраща ефективність виявлення досягається, коли модуль C2f між шарами B4 і B5 базової моделі замінюється на модуль BiFormer. Порівняно з базовою моделлю, mAP0.5 покращується на 0.5%. При додаванні блоку BiFormer модель може краще зосередитися на важливій інформації у вхідних характеристиках.

Таблиця 3.7 – Експериментальні результати після введення BiFormer блоку на різні шари хребта (backbone) мережі

Модель	Точність (P), %	Повнота (R), %	mAP0.5, %	mAP0.5:0.95, %
Baseline	50.7	38.7	40.0	23.6
+B3-BiFormer	49.7	38.6	39.7	23.5
+B4-BiFormer	50.6	39.2	40.5	24.1
+B3-BiFormer+	50.3	39.1	40.2	24.0
B4-BiFormer	50.3	39.1	40.2	24.0

3.4.4. Експерименти з порівнянням впровадження різних стратегій покращення

При порівнянні двох методів YOLO v9 та YOLOv9-P було отримано результати, що наведені в таблиці 3.8. Експеримент було проведено на 114 зображеннях виходячи з обмежень до обчислювальних потужностей.

Таблиця 3.8 – Експериментальні результати технології YOLOv9 та YOLOv9-P на наборі зображень у кількості 114

Метод	Кількість зображень	Точність (P), %	Повнота (R), %	mAP50, %	mAP0.5:0.95, %
YOLO v9	114	0.0011	0.0795	0.00141	0.00436
YOLO v9 P	114	0.00101	0.0487	0.0025	0.000865

Для перевірки ефективності кожної стратегії покращення, запропонованої в цій роботі, ми проведено моделювання експериментів з порівнянням на базовій моделі з використанням набору даних TAIVD, і результати експериментів наведено в таблиці 3.9.

B2-PFNB і B1-PFNB у таблиці 3.9 вказують на використання шарів виявлення на основі модулів PFNB, які об'єднують дрібні особливості шарів B2 і B1, відповідно. Знак «+» вказує на те, що було використано цю покращену стратегію.

Дані експериментальні результати, що наведені в таблиці 3.8 показують, що кожна стратегія покращення різною мірою покращила ефективність виявлення, коли її було застосовано до базової моделі. WIoU v3 вводиться в блок регресійних втрат прогнозу. WIoU v3 покращує здатність моделі до локалізації, використовуючи більш розумну стратегію розподілу вибірок, що призводить до покращення mAP50 на 0,7%. BiFormer було впроваджено в магістральну мережу, замінивши модуль C2f в оригінальній моделі. Ефективний механізм уваги в BiFormer покращує увагу до ключової інформації на карті об'єктів, що збільшує mAP50 на 0,5%. Відносно проста структура модуля BiFormer зменшує розмір моделі на 0,4 МБ на невеликому наборі даних, а кількість параметрів – приблизно на 0,2 М (млн). До базової моделі було додано шар детектування B2-PFNB, що збільшує mAP на 4,3%.

Таблиця 3.9 – Результати виявлення після впровадження різних стратегій покращення

	W I o U v 3	Bi Fo rm er	B1 - PF N B	B2 - PF N B	Точн ість (P), %	Пов нот а (R), %	mA P 0.5, %	mAP 0.5:0 .95, %	Розмір моделі, МВ	Час виявле ння, мс	Пар аме тр, 10 ⁶
YOLO v9 P					50.9	38.2	39.3	23.5	22.5	7.7	11.1
	+				50.7	38.7	40.0	23	22.5	7.2	11.1
	+	+			50.6	39.2	40.5	24.1	22.1	7.8	10.9
	+	+	+		55.8	42.8	44.8	27.2	21.6	11.2	10.6
	+	+	+	+	54.4	46.5	47.0	29.5	21.5	19.5	10.3

Результати покращеного алгоритму YOLO v9 P на наборі TAIVD та власного набору даних наведено в таблиці 3.10.

Таблиця 3.10 – Результати покращеного алгоритму YOLO v9 P на наборі TAIVD та власного dataset

	Точн ість (P), %	Повно та (R), %	mAP 0.5, %	mAP0. 5:0.95, %	Розмір моделі, МВ	Час виявле ння, мс	Парам етр, 10 ⁶
TAIVD	52.6	46.5	46.8	28.9	22.8	22.3	10.3
Власний набір даних	54.4	46.5	47.0	29.5	21.5	19.5	10.3

Дані експериментальні результати, що наведені в таблиці 3.9 показують, що тестування на власному наборі даних більш точні та є швидкими при однакової кількості параметрів.

3.5 Висновки до розділу 3

У третьому розділі в результаті проведеного дослідження проведено обґрунтоване розроблення програмного забезпечення при врахуванні результатів порівняння з існуючими програмними рішеннями.

Проведено огляд існуючих наборів даних (зокрема, Kitti та TAIVD) для подальшої апробації алгоритмів та обговорюється питання анотування даних та формування власного набору даних (dataset, датасет). Отримані результати спостережень з побудованого власного набору даних мають перевагу за розміром моделі на 6 % відносно датасету TAIVD, який може бути доречним для використання за рахунок вміщення зображень з БпЛА.

Проведено моделювання порівняльних експериментів на YOLO v9 на невеликому наборі даних при використанні різних видів функцій втрат та збереженні інших умов навчання незмінними. Результати моделювання підтвердили вибір правильної стратегії щодо використання функції регресійних втрат WIoU v3 для побудованої технології.

Проведено моделювання експериментів при додаванні до базової моделі блоків детектування групи PFNB. Результати моделювання показали збільшення точності mAP навіть на невеликому наборі даних. При одночасному використанні блоків PFNB в побудованій технології розмір моделі і кількість параметрів зменшується.

Запропоновано та побудовано технологію YOLO v9 P для вирішення задачі розпізнавання та відстеження об'єктів в режимі реального часу, що надає більш точні результати при меншій обчислювальній складності у

порівнянні з YOLO v9 і дозволяє використовувати її в обмежених за ресурсами системах.

ВИСНОВКИ

Сукупність отриманих в дисертаційному дослідженні результатів вирішує актуальне науково-технічне завдання розроблення технології підвищення точності розпізнавання та відстежування об'єктів у режимі реального часу.

Зазначене наукове завдання має істотне значення для розвитку технологій глибокого навчання, алгоритмів та методів покращення роботи програмних рішень при використанні на борту БпЛА. Враховуючи стрімкий розвиток інформаційних технологій, постійне оновлення вже існуючих рішень та обмеженість доступу до інформаційних ресурсів наукових публікацій, можна визначати результати досліджень такими, що мають високу значущість.

В дисертаційному дослідженні отримані наступні основні результати:

1. Вперше розроблено архітектурне рішення побудови нейронної згорткової мережі задачі виявлення, розпізнавання та відстеження об'єктів в режимі реального часу, що відрізняється від існуючого рішення тим, що використовує більшу кількість блоків розпізнавання об'єктів різного розміру, яке є оптимізоване для задач конкретної предметної області.

2. Вперше обґрунтовано можливість використання в розробленій технології PFNB-блоку, який базується на архітектурному рішенні Faster-Net, що використовує багатомасштабну мережу об'єднання ознак для демонстрації покращеної точності розпізнавання у порівнянні з базовою.

3. Вперше сформований власний набір даних для апробації розробленої технології починаючи з етапу розпізнавання об'єктів у відеопотоці, який включає об'єкти різного масштабу визначеної предметної області, що підтверджує ефективність розробленої моделі.

4. Вперше запропоновано архітектуру кроссплатформної бібліотеки для реалізації технології виявлення, розпізнавання та відстеження об'єктів, яка є п'ятимасштабною структурою і містить механізм уваги ViFormer з малою

обчислювальною потужністю (зменшує розмір моделі на 0,4 МБ на невеликому наборі даних), що дозволяє покращити точність виявлення малих об'єктів та покращує увагу до ключової інформації на карті об'єктів і збільшує mAP50 на 0,5%.

5. Вперше проведено моделювання порівняльних експериментів на YOLO v9 на невеликому наборі даних, які відрізняються використанням різних видів функцій втрат при зберіганні інших умов навчання незмінними, що показало використання функції регресійних втрат WIoU v3 найефективнішою для побудованої моделі і значення mAP моделі при використанні WIoU v3 на 0,7% вище, ніж при використанні CIoU.

6. Вперше проведено моделювання експериментів при додаванні до базової моделі блоків детектування групи PFNB, які об'єднують дрібні особливості шарів нейронної згорткової мережі, що збільшує на невеликому наборі даних значення mAP на 4,3% та при їх одночасному використанні розмір моделі і кількість параметрів зменшується зменшується на 5% (1 MB) і кількість параметрів зменшується більше ніж на 7,2% ($0,8 * 10^6$).

7. Вперше проведено моделювання експериментів на покращеній моделі YOLO v9 P, яка відрізняється від базової моделі YOLO v9 функцією втрат, методом злиття та модифікованою архітектурою блоку розпізнавання, що на невеликому наборі даних дозволило отримати покращення значень mAP на 7,7% і AP від 2,5% до 14,1%.

8. Практичне використання розробленої технології підтверджено експериментальним впровадженням в ТОВ «КАНЬОН ІНЖИНІРИНГ», зокрема у процес розробки проєкту «FridgeEye», що дозволяє проводити розпізнавання об'єктів у реальному часі та використовувати в інтелектуальних системах з обмеженими ресурсами.

Таким чином, усі поставлені завдання у даному науковому дослідженні повністю виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Zhenxiong W., Kai K., Xueying H., Huanming H., Jianghai L., Lang C. Computational simulation study on disturbance of six-rotor UAVs due to ammunition launch. Journal of Physics: Conference Series. 2023. Vol. 2478, article no. 102022. P. 1–19. DOI: 10.1088/1742-6596/2478/10/102022
- 2 Федій О.Д., Божуха Л.М. Про підходи визначення місцезнаходження об'єктів. Науковий журнал «Математичне моделювання». Кам'янське, 2021. – Вип. 2(45). – С.39-46. Режим доступу до ресурсу: [https://doi.org/10.31319/2519-8106.2\(45\)2021.246874](https://doi.org/10.31319/2519-8106.2(45)2021.246874) [Фахове видання України категорії Б]
- 3 Сизоненко О.Д., Божуха Л.М. Підвищення точності геолокації об'єкта на цифровому зображенні при використанні комбінованих технологій аналізу даних. Науковий журнал «Актуальні проблеми автоматизації та інформаційних технологій». Дніпро, 2022. – Т.26.– С.103-109. Режим доступу до ресурсу: <http://dx.doi.org/10.15421/432213> [Фахове видання України категорії Б]
- 4 Виявлення місцезнаходження об'єктів за допомогою GIS / Сизоненко О.Д., Божуха Л.М // XX Міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем”, м. Дніпро (23 – 25 листопада 2022 р.), 2022, с.178
- 5 Про алгоритми позиціювання об'єктів в локальній мережі / Федій О.Д., Божуха Л.М // XIX Міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем”, м. Дніпро (17 – 19 листопада 2021 р.), 2021, с.201
- 6 Методи прив'язки зображення до геолокації / Сизоненко О.Д., Божуха Л.М. // Всеукраїнська науково-методична конференція “Проблеми математичного моделювання”, м.Кам'янське (25-27 травня 2022 р.), 2022, с.84

7 Довідкове посилання: Діаграма з прогнозованим зростанням ринку комерційних дронів. Режим доступу до ресурсу: <https://www.statista.com/chart/17201/commercial-drones-projected-growth/>

8 Виявлення місцезнаходження бпла за допомогою зіставлення зображень з використанням ключових точок /Сизоненко О. Д., Божуха Л.М. // XXI Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро (22 – 24 листопада 2023 р.), 2023, с.266-267

9 Yang, Shao-Yu & Cheng, Hsu-Yung & Yu, Chih-Chang. Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks. Electronics. 2023. 12. 4928. DOI: 10.3390/electronics12244928 Режим доступу до ресурсу: https://www.researchgate.net/publication/376348877_Real-Time_Object_Detection_and_Tracking_for_Unmanned_Aerial_Vehicles_Based_on_Convolutional_Neural_Networks

10 Viola, Paul A. and Michael Jones. Robust Real-time Object Detection. International Journal of Computer Vision. 2001. DOI: 10.1109/ICCV.2001.937709 Режим доступу до ресурсу: https://www.researchgate.net/publication/221111859_Robust_Real-Time_Face_Detection

11 Сизоненко О.Д., Божуха Л.М. Методи локалізації об'єктів на основі зображень із використанням комбінації алгоритмів та багатопоточної зв'язки Faster R-CNN. // Актуальні проблеми автоматизації та інформаційних технологій. – Дніпро: Ліра, 2023. Т.27., с. 164-177. Режим доступу до ресурсу: <http://dx.doi.org/10.15421/432316> [Фахове видання України категорії Б]

12 Експериментальні результати встановлення геолокації об'єкта при використанні мережі виявлення об'єктів Faster R-CNN / Сизоненко О.Д., Божуха Л.М. // Наукова конференція за підсумками науково-дослідної роботи ДНУ ім.О.Гончара за 2022 рік. – Д.: ДНУ, 2022

- 13 Girshick, R., Donahue, J., Darrell, T., & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, p. 580-587. DOI:10.1109/CVPR.2014.81.
<https://pdfs.semanticscholar.org/2ec7/3ae35fe874bdc3ccc143276801d4f57b7f08.pdf>
- 14 Han, S., Shen, W., & Liu, Z. Deep drone: object detection and tracking for smart drones on embedded system. 2017. URL: <https://api.semanticscholar.org/CorpusID:53981974>
- 15 Rohan, A., Rabah, M., & Kim, S. H. (2019). Convolutional neural networkbased realtime object detection and tracking for parrot AR drone 2. in IEEE Access, vol. 7, pp. 69575-69584, 2019, DOI: <https://doi.org/10.1109/ACCESS.2019.2919332>.
- 16 Ramachandran, Anitha & Sangaiah, Arun. A Review on Object Detection in Unmanned Aerial Vehicle Surveillance. International Journal of Cognitive Computing in Engineering. 2. 10.1016/j.ijcce. 2021.11.005.
- 17 https://www.mdpi.com/remotesensing/remotesensing-12-02501/article_deploy/html/images/remotesensing-12-02501-g001.png
- 18 Liu, C.Y.; Wu, Y.Q.; Liu, J.J.; Han, J.M. MTI-YOLO: A Light-Weight and Real-Time Deep Neural Network for Insulator Detection in Complex Aerial Images. Energies 2021, 14, 1426
- 19 Sahin, O.; Ozer, S. YOLO Drone: Improved YOLO Architecture for Object Detection in Drone Images. In Proceedings of the 44th International Conference on Telecommunications and Signal Processing (TSP), Virtual, 26–28 July 2021; pp. 361–365
- 20 Junos, M.H.; Khairuddin, A.S.M.; Thannirmalai, S.; Dahari, M. Automatic detection of oil palm fruits from UAV images using animproved YOLO model. Vis. Comput. 2022,38, 2341–2355

- 21 Guo, J.; Xie, J.; Yuan, J.; Jiang, Y.; Lu, S. Fault Identification of Transmission Line Shockproof Hammer Based on Improved YOLO V4. In Proceedings of the 2021 International Conference on Intelligent Computing, Automation and Applications (ICAA), Nanjing, China, 25–27 June 2021; pp. 826–833
- 22 Cheng, Y. Detection of Power Line Insulator Based on Enhanced YOLO Model. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2022, Dalian, China, 14–16 April 2022; pp. 626–632
- 23 Ding, W.; Zhang, L. Building Detection in Remote Sensing Image Based on Improved YOLOV5. In Proceedings of the 17th International Conference on Computational Intelligence and Security, CIS 2021, Chengdu, China, 19–22 November 2021; pp. 133–136.8
- 24 Wang, X.W.; Zhao, Q.Z.; Jiang, P.; Zheng, Y.C.; Yuan, L.M.Z.; Yuan, P.L. LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest. *Comput. Electron. Agric.* 2022, 198, 107035.
- 25 Liu, Y.; Shi, G.; Li, Y.; Zhao, Z. M-YOLO based Detection and Recognition of Highway Surface Oil Filling with Unmanned aerial vehicle. In Proceedings of the 7th International Conference on Intelligent Computing and Signal Processing, ICSP 2022, Xi'an, China, 15–17 April 2022; pp. 1884–1887
- 26 Zhang, R.; Wen, C.B. SOD-YOLO: A Small Target Defect Detection Algorithm for Wind Turbine Blades Based on Improved YOLOv5. *Adv. Theory Simul.* 2022, 5, 2100631
- 27 Офіційний сайт з документацією бібліотеки ultralytics
URL: <https://docs.ultralytics.com/>
- 28 Laura Leal-Taixé. Multiple object tracking with context awareness
URL: <https://dblp.org/rec/journals/corr/Leal-Taixe14>

- 29 Kamate, S., & Yilmazer, N. Application of object detection and tracking techniques for unmanned aerial vehicles. Elsevier, 61, 436–441. 10.1016/j.procs. 2015.09.18.
- 30 Chu, Peng et al. TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2021. p. 4859-4869.
- 31 Zhang, Yifu, Pei Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu and Xinggang Wang. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. European Conference on Computer Vision. 2021. ArXiv: 2110.06864. DOI: <https://doi.org/10.48550/arXiv.2110.06864>
- 32 Сизоненко О.Д., Божуха Л.М. Порівняння YOLO V5 та Faster R-CNN для виявлення об'єктів на зображенні в потоковому режимі. Системні технології. Дніпро, 2024. – 1(150) – С. 51-60. Режим доступу до ресурсу: <https://doi.org/10.34185/1562-9945-1-150-2024-05> [Фахове видання України категорії Б]
- 33 Redmon, Joseph and Ali Farhadi. YOLOv3: An Incremental Improvement. 2018 ArXiv: 1804.02767 DOI: <https://doi.org/10.48550/arXiv.1804.02767>
- 34 Wang C-Y, Liao H-YM, Yeh I-H, Wu Y-H, Chen P-Y, Hsieh J-W. CSPNet: A new Backbone that can Enhance Learning Capability of CNN. 2019; Available from: <http://arxiv.org/abs/1911.11929>
- 35 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. [CrossRef] [PubMed]
- 36 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

- 37 Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. arXiv 2020, arXiv:2006.04388
- 38 Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000
- 39 Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In Proceedings of the 2021 IEEE International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3490–3499.
- 40 Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. BiFormer: Vision Transformer with Bi-Level Routing Attention. arXiv, arXiv:2303.08810.
- 41 Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. Neurocomputing 2022, 506, 146–157. [CrossRef]
- 42 Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. arXiv 2022, arXiv:2205.12740.
- 43 Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv, arXiv:2301.10051.
- 44 Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. IEEE Trans. Pattern Anal. Mach. Intell. 2021, 44, 7380–7399. [CrossRef] [PubMed]
- 45 Chen, J.; Kao, S.-H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023.
- 46 Kupyn, Orest, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin and Jiri Matas. DeblurGAN: Blind Motion Deblurring Using Conditional

Adversarial Networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition . 2018. arXiv: 1711.07064. DOI: <https://doi.org/10.48550/arXiv.1711.07064>

47 Bemposta Rosende, S.; Ghisler, S.; Fernández-Andrés, J.; Sánchez-Soriano, J. Dataset: Traffic Images Captured from UAVs for Use in Training Machine Vision Algorithms for Traffic Management. 2022 . DOI: <https://doi.org/10.3390/data7050053>

ДОДАТОК А Список публікацій здобувача

Статті у наукових фахових виданнях України:

1. Федій О.Д., Божуха Л.М. Про підходи визначення місцезнаходження об'єктів. *Науковий журнал «Математичне моделювання»*. 2021. Вип. 2(45). С. 39-46. DOI: [https://doi.org/10.31319/2519-8106.2\(45\)2021.246874](https://doi.org/10.31319/2519-8106.2(45)2021.246874) URL: <http://matmod.dstu.dp.ua/article/view/246874> **(фахове видання категорії Б).**

2. Сизоненко О.Д., Божуха Л.М. Підвищення точності геолокації об'єкта на цифровому зображенні при використанні комбінованих технологій аналізу даних. *Науковий журнал «Актуальні проблеми автоматизації та інформаційних технологій»*. 2022. Т.26. С. 103-109. DOI: <http://dx.doi.org/10.15421/432213> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/221> **(фахове видання категорії Б).**

3. Сизоненко О.Д., Божуха Л.М. Методи локалізації об'єктів на основі зображень із використанням комбінації алгоритмів та багатопоточної зв'язки Faster R-CNN. *Актуальні проблеми автоматизації та інформаційних технологій*. 2023. Т.27. С. 164-177. DOI: <http://dx.doi.org/10.15421/432316> URL: <https://actualproblems.dp.ua/index.php/APAIT/article/view/241> **(фахове видання категорії Б).**

4. Сизоненко О.Д., Божуха Л.М. Порівняння YOLO V5 та Faster R-CNN для виявлення об'єктів на зображенні в потоковому режимі. *Системні технології*. 2024. 1(150). С. 51-60. DOI: <https://doi.org/10.34185/1562-9945-1-150-2024-05> URL: <https://journals.nmetau.edu.ua/index.php/st/article/view/1523> **(фахове видання категорії Б).**

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Сизоненко О. Д., Божуха Л.М. Виявлення місцезнаходження бпла за допомогою зіставлення зображень з використанням ключових точок. XXI Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем»: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2023 рік. Дніпро, 2023, С. 266-267, URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2023/11/mpzis-2023.pdf>.

6. Сизоненко О.Д., Божуха Л.М. Виявлення місцезнаходження об'єктів за допомогою GIS. XX Міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем”: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2022 рік. Дніпро, 2022, С. 178, URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf>.

7. Федій О.Д., Божуха Л.М. Про алгоритми позиціювання об'єктів в локальній мережі. XIX Міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем”: тези доповідей наукової конференції за підсумками науково-дослідної роботи ДНУ за 2021 рік. Дніпро, 2021, С. 201, URL: http://mpzis.dnu.dp.ua/wp-content/uploads/2021/11/mpzis_2021.pdf.

8. Сизоненко О.Д., Божуха Л.М. Методи прив'язки зображення до геолокації. Всеукраїнська науково-методична конференція “Проблеми математичного моделювання”: тези доповідей Всеукраїнської науково-методичної конференції за 2022 рік. Кам'янське, 2022, С. 84, URL: https://www.dstu.dp.ua/uni/downloads/zbirka_konf_pm.pdf.

9. Сизоненко О.Д., Божуха Л.М. Експериментальні результати встановлення геолокації об'єкта при використанні мережі виявлення об'єктів Faster R-CNN. Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022): тези доповідей XX міжнародної науково-практичної

конференції, Дніпро, 2022, виступ є, без публікації, URL:
https://www.dnu.dp.ua/docs/ndc/2023/Ost_var_programa.pdf.

ДОДАТОК Б Акт впровадження

«ЗАТВЕРДЖУЮ»
Директор
ТОВ «КАНЬОН ІНЖИНІРІНГ»
Микола БОГУН
2024 р.
43032376
УКРАЇНА, м. Дніпро

АКТ

впровадження результатів дисертаційної роботи

Сизоненко Олександри Дмитрівни

Даним актом представники ТОВ «КАНЬОН ІНЖИНІРІНГ» в особі Директора Миколи Богуна, засвідчує, що результати дисертаційної роботи аспірантки кафедри математичного забезпечення ЕОМ факультету прикладної математики «Дніпровського національного університету імені Олеся Гончара» Сизоненко Олександри Дмитрівни за темою «Розроблення технології та програмних засобів виявлення та розпізнавання об'єктів у режимі реального часу» використовується у процесі розробки проєкту «FridgeEye»:

- Запропонований алгоритм YOLO v9 P дозволяє проводити розпізнавання об'єктів у реальному часі;
- Запропонований та побудований алгоритм видає більш точні результати, затрачуючи менше ресурсів у порівнянні з YOLO v9, що дозволяє його використовувати в системах, що є обмеженими за ресурсами.

Директор

ТОВ «КАНЬОН ІНЖИНІРІНГ»

Микола БОГУН