

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

«На правах рукопису»
УДК 004.932.72:629.584(043.3)

До захисту допущено:
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«___» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Системи і методи штучного інтелекту»

зі спеціальності 122 «Комп'ютерні науки»

**на тему: «Інтелектуальна обробка зображень на автономних підводних
апаратах»**

Виконав:

студент II курсу, групи КІ-31мп
Савченко Михайло Володимирович

Науковий керівник:

професор кафедри ШІ, д.т.н., професор,
Синеглазов Віктор Михайлович

Рецензент:

доцент кафедри АКІК КАІ, к.т.н.,
Василенко Микола Павлович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«29» серпня 2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Савченку Михайлу Володимировичу

1. Тема дисертації «Інтелектуальна обробка зображень на автономних підводних апаратах», науковий керівник дисертації Синєглазов Віктор Михайлович, д.т.н, професор, затверджені наказом по НН ІПСА від «07» листопада 2024 р. № 5001-с
2. Термін подання студентом дисертації «13» грудня 2024 року
3. Об'єкт дослідження: методи комп'ютерного зору та детектування об'єктів
4. Вихідні дані: набори підводних зображень URPC2019 та UTDAC2020
5. Перелік завдань, які потрібно розробити:
 - 1) Здійснити огляд технічної літератури за тематикою роботи та існуючих інтелектуальних систем
 - 2) Дослідити предметну область та існуючі системи
 - 3) Проаналізувати актуальність обраної теми
 - 4) Здійснити аналіз та обробку навчальних вибірок, вирішити проблеми балансування вибірки та покращення якості знімків
 - 5) Розробити методологію інтелектуальної обробки зображень на автономних підводних апаратах
 - 6) Провести експерименти та здійснити аналіз результатів
 - 7) Проаналізувати ринкові можливості запуску стартап-проекту
 - 8) Розробити висновки з проведених досліджень
 - 9) Оформити пояснювальну записку магістерської дисертації

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація, веб-інтерфейс для демонстрації програмного забезпечення

7. Орієнтовний перелік публікацій:

- 1) V. Sineglazov та M. Savchenko, “Comprehensive Framework for Underwater Object Detection Based on Improved YOLOv8”, Electron. Control Syst., т. 1, № 79, с. 9–15, берез. 2024. Дата звернення: 30 серп. 2024. [Онлайн]. Доступно: <https://doi.org/10.18372/1990-5548.79.18429>
- 2) Заплановано апробувати роботу «Efficient small and overlapping target detection in underwater images» на конференції ProfIT AI 2024 у вересні 2024 року.
- 3) Заплановано апробувати роботу «Optimal size reduction methodology for YOLO-based underwater object detectors based on knowledge distillation» на конференції IT&T 2024 у листопаді 2024 року.

8. Дата видачі завдання «30» серпня 2024 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Опрацювання літератури за темою роботи	02.09.2024 – 15.09.2024	Виконано
2	Підготовка першого розділу	16.09.2024 – 24.09.2024	Виконано
3	Підготовка другого розділу	25.09.2024 – 04.10.2024	Виконано
4	Розробка алгоритмічного та програмного забезпечення інтелектуальної системи	05.10.2024 – 15.11.2024	Виконано
5	Підготовка третього розділу	16.11.2024 – 27.11.2024	Виконано
6	Підготовка четвертого розділу	28.11.2024 – 08.12.2024	Виконано
8	Оформлення пояснювальної записки	09.12.2024 – 13.12.2024	Виконано

Студент

Михайло САВЧЕНКО

Науковий керівник

Віктор СИНЄГЛАЗОВ

РЕФЕРАТ

Магістерська дисертація: 103 с., 38 рис., 19 табл., 44 посилання, додаток.

ДЕТЕКТУВАННЯ ПІДВОДНИХ ОБ'ЄКТІВ, АВТОНОМНІ ПІДВОДНІ АПАРАТИ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, YOLO

Об'єкт дослідження – технології комп'ютерного зору.

Предмет дослідження – інтелектуальна обробка зображень, знятих у підводному середовищі.

Мета роботи – створення інтелектуальної системи, здатної ефективно вирішувати задачу розпізнавання підводних об'єктів при низькій обчислювальній складності.

Результатами дослідження є створення авторської методології, яка включає метод попередньої обробки зображень для зменшення впливу оптичних перешкод у підводному середовищі; метод структурно-параметричного синтезу нейронної мережі для розпізнавання об'єктів на підводних знімках; метод стиснення нейронної мережі для забезпечення роботи в умовах обмеженої потужності та автономності.

Практичне значення роботи полягає в можливості побудови нейронної мережі детектування об'єктів на основі поданої методології та її розміщення на окремому апаратному модулі для автономного підводного апарату, який дозволить йому виконувати прикладні задачі, що базуються на локалізації та класифікації об'єктів інтересу.

Основні положення роботи викладено в роботі В. Синєглазова, М. Савченка і В. Литвиненка "Efficient small and overlapping target detection in underwater images" та представлено на 4th Int. Workshop IT-professionals Artif. Intell. (ProfIT AI 2024) у вересні 2024 року.

ABSTRACT

Master's thesis: 103 p., 38 figures, 19 tables, 44 references, appendix.

KEYWORD 1, UNDERWATER OBJECT DETECTION, AUTONOMOUS UNDERWATER VEHICLES, MACHINE LEARNING, DEEP LEARNING, NEURAL NETWORKS, YOLO

The object of study are computer vision technologies.

The subject of study is the intellectual processing of images taken in the underwater environment.

The purpose of the study is to create an intelligent system capable of effectively solving the problem of recognizing underwater objects at low computational complexity.

The results are the creation of the author's methodology, which includes a method of image preprocessing to reduce the impact of optical interference in the underwater environment; a method of structural and parametric synthesis of a neural network for recognizing objects in underwater images; a method of neural network compression to ensure operation in conditions of limited computational power and autonomous work capabilities.

The practical significance of the work lies in the possibility of building a neural network for object detection based on the presented methodology and its placement on a separate hardware module for an autonomous underwater vehicle, which will allow it to perform applied tasks based on the localization and classification of the objects of interest.

Main points of the work have been introduced in the paper by V. Sineglazov, M. Savchenko and V. Lytvynenko named "Efficient small and overlapping target detection in underwater images," presented on 4th Int. Workshop IT-professionals Artif. Intell. (ProfIT AI 2024) in September 2024.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТУ ДОСЛІДЖЕННЯ ТА СУЧАСНИХ МЕТОДІВ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ	10
1.1 Поняття автономних підводних апаратів	10
1.2 Огляд існуючих методів детектування об'єктів	14
1.2.1 Методи на основі машинного навчання	15
1.2.2 Методи на основі нейронних мереж	18
1.3 ПРОБЛЕМАТИКА СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ДЛЯ АВТОНОМНИХ ПІДВОДНИХ АПАРАТІВ	24
1.4 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ	29
Висновки до розділу 1	30
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ НЕЙРОННОЇ МЕРЕЖІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ	32
2.1 ПОПЕРЕДНЯ ОБРОБКА ЗОБРАЖЕНЬ	32
2.2 МЕТОДИ YOLO ДЛЯ ЗАДАЧ ДЕТЕКТУВАННЯ.....	40
2.2.1 Загальна характеристика методів YOLO.....	40
2.2.2 Алгоритмічні основи YOLO	42
2.2.3 Топологія нейронних мереж на основі YOLO	45
2.2.4 Теоретичні аспекти навчання мережі YOLO	49
2.3 МЕТОДИ КОМПРЕСІЇ НЕЙРОННИХ МЕРЕЖ НА БАЗІ YOLO	51
2.4 ГІБРИДНІ МОДЕЛІ ДЕТЕКТУВАННЯ ПІДВОДНИХ ОБ'ЄКТІВ	52
2.4.1 Загальна характеристика.....	52
2.4.2 Огляд робіт	54
Висновки до розділу 2	56
РОЗДІЛ 3 ІНТЕЛЕКТУАЛЬНА СИСТЕМА ОБРОБКИ ЗОБРАЖЕНЬ НА АВТОНОМНИХ ПІДВОДНИХ АПАРАТАХ	58

	7
3.1 МЕТОДОЛОГІЯ ДЕТЕКТУВАННЯ ПІДВОДНИХ ОБ’ЄКТІВ.....	59
3.1.1 Вибір методів попередньої обробки	59
3.1.2 Визначення топології нейронної мережі	61
3.2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ	65
3.3 НАВЧАЛЬНІ ВИБІРКИ	68
3.4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ.....	70
3.4.1 Метрики для оцінки ефективності мережі	70
3.4.2 Середовище розробки, гіперпараметри і оптимізатор	71
3.4.3 Результати експериментів.....	71
3.5 СТРУКТУРА ТА ІНТЕРФЕЙС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	74
Висновки до розділу 3.....	77
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ СИСТЕМИ ДЕТЕКТУВАННЯ	
.....	78
4.1 ОПИС ІДЕЇ ПРОЄКТУ	78
4.2 ТЕХНОЛОГІЧНИЙ АУДИТ ПРОЄКТУ	80
4.3 АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ ЗАПУСКУ ПРОЄКТУ	82
4.4 РОЗРОБЛЕННЯ РИНКОВОЇ СТРАТЕГІЇ ПРОЄКТУ	89
Висновки до розділу 4.....	91
ВИСНОВКИ	92
ПЕРЕЛІК ПОСИЛАНЬ	94
ДОДАТОК А ЛІСТИНГ КЛЮЧОВИХ СКЛАДНИКІВ ПРОГРАМИ	100

ВСТУП

Штучний інтелект активно розвивається, зокрема у сфері комп'ютерного зору. Методи детектування, відслідковування та сегментації об'єктів на основі нейронних мереж дозволили автоматизувати низку процесів на виробництвах, у сфері безпеки та повсякденному житті. Щодня з'являються нові алгоритми, програмне забезпечення та наукові роботи, що значно впливають на розвиток технологій штучного інтелекту.

Однією з перспективних галузей застосування ШІ є автономні підводні апарати — безпілотні пристрої, здатні виконувати завдання на значних відстанях і глибинах від точки запуску. Завдяки ШІ, такі апарати можуть виконувати комплексні завдання: від автономної навігації до пошуку об'єктів, їх розпізнавання та інтелектуальної оцінки середовища.

Дана магістерська робота присвячена дослідженню методів пошуку, локалізації та класифікації об'єктів безпосередньо на автономних підводних апаратах. **Об'єктом** дослідження є технології комп'ютерного зору, а **предметом** — інтелектуальна обробка зображень, знятих у підводному середовищі. **Актуальність** цієї роботи обумовлена необхідністю розробити інтелектуальну систему для обробки оптичних зображень, яка б могла функціонувати в умовах обмежених обчислювальних ресурсів та адаптуватися до специфіки підводного середовища.

Метою цієї роботи є створення інтелектуальної системи, здатної ефективно вирішувати задачу розпізнавання підводних об'єктів. У межах дослідження виконано огляд існуючих методів розпізнавання об'єктів, проаналізовано їх обмеження, а також запропоновано підходи до подолання цих проблем, а саме: знижена ефективність при роботі з зображеннями з проблемами контрастності та недостатнього освітлення, хибно-негативні визначення об'єктів, що розташовані щільно та перекриваються, висока обчислювальна складність гібридної нейронної мережі.

Результатом дослідження стала розробка авторської методології, яка включає метод попередньої обробки зображень для зменшення впливу оптичних перешкод у підводному середовищі; метод структурно-параметричного синтезу нейронної мережі для розпізнавання об'єктів на підводних знімках; метод стиснення нейронної мережі для забезпечення роботи в умовах обмеженої потужності та автономності.

Створена методологія і програмне забезпечення спрямовані на підвищення ефективності роботи автономних підводних апаратів в умовах складного підводного середовища, що дозволить використовувати їх у низці прикладних задач.

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТУ ДОСЛІДЖЕННЯ ТА СУЧАСНИХ МЕТОДІВ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ

1.1 Поняття автономних підводних апаратів

Автономний підводний апарат (скор. — АПА, англ. — Autonomous Underwater Vehicle, AUV) — це автономний робот, який працює під водою без керування людиною. На відміну від інших роботизованих підводних суден, таких як дистанційно керовані апарати (англ. — Remotely Operated Vehicles, ROV), автономні апарати мають автономну систему навігації та контролю. Такий пристрій може самостійно орієнтуватися в середовищі, виконувати місії та приймати рішення на основі попередньо закладених алгоритмів і даних, отриманих з камер та сенсорів в режимі реального. Важливість такої автономії полягає в можливості проводити дослідження на значних глибинах і відстанях від точки запуску. Це дозволяє виконувати завдання, які недоступні або надто небезпечні для пілотованих місій.

Розробка автономних підводних апаратів є важливим напрямом у сфері морської інженерії та робототехніки. Вони розробляються з використанням передових технологій комп'ютерного зору, машинного навчання та навігаційних систем, що дозволяє забезпечити стабільність роботи навіть у важкодоступних умовах. Такий набір унікальних можливостей привернув увагу дослідників одразу в низці сфер. Розглянемо приклади галузей науки та бізнесу, в яких застосовуються автономні підводні апарати.

У **військовій сфері** АПА часто застосовують для завдань військової розвідки, в тому числі для моніторингу та дослідження морських шляхів, виявлення, оцінку ступеню небезпеки та нейтралізацію підводних мін (**рис. 1.1**), а також для збору даних про активності інших країн в підводних середовищах. Деякі військові апарати обладнані спеціальні технології для роботи на великих глибинах, що дозволяє проводити стратегічно важливі операції на недоступних для ворога рівнях.



Рисунок 1.1 — Військовий автономний підводний апарат для задач розмінування

АПА є цінним інструментом для **наукової сфери**, зокрема, для морських біологів, океанографів та екологів. Автономні апарати дозволяють проводити довготривалі спостереження за морським дном, збирати проби води та ґрунту, відслідковувати поведінку морських організмів та проводити оцінку біорізноманіття в підводному середовищі. Їх в науковій сфері також підсилюється здатністю до дослідження океанічних глибин і зон, де тиск та складні умови не дозволяють використовувати традиційні методи дослідження з участю людини.

Іншим прикладом успішного застосування автономних підводних апаратів є **енергетичний сектор**. В нафтовій та газовій промисловості АПА використовуються для обстеження морського дна та обслуговування підводних трубопроводів і платформ. Їх застосування значно знижує ризики для персоналу, дозволяючи проводити технічне обслуговування та моніторинг інфраструктури на великих глибинах, в умовах низької видимості та

¹ Джерело зображення — <https://commons.wikimedia.org/wiki/File:MiniU.jpg>

забрудненості середовища. Завдяки можливості точного маневрування та автономного виконання завдань, АПА стали ключовим елементом для забезпечення безперервної та безпечної роботи нафтових платформ.

Також АПА застосовуються **рятувальних операцій та дослідження затонулих або втрачених об'єктів**. Їх застосовують для пошуку затонулих суден, літаків та інших об'єктів, оскільки вони здатні працювати у важкодоступних умовах, де використання іншого обладнання є недоцільним фінансово або через ризик його втрати чи пошкодження. Крім того, можливість тривалого автономного функціонування дозволяє АПА виконувати дослідження в зонах високого ризику для людини. Зокрема, АПА часто використовуються для розслідування авіакатастроф та пошуку уламків, наприклад, американський підводний дрон Bluefin-21 (рис. 1.2) успішно застосовувався у пошукових операціях зниклого борту 370 Малазійських авіаліній.



Рисунок 1.2 — Автономний підводний апарат Bluefin-21, що використовувався для пошуку втрачених повітряних суден

² Джерело зображення — https://www.researchgate.net/figure/The-PAUL-a-Bluefin-21-AUV-equipped-with-physical-and-biogeochemical-oceanographic_fig6_343597772

Будова АПА орієнтована на забезпечення високої маневреності, захищеності й ефективності в умовах підвищеного тиску та низької видимості. Ключові елементи конструкції включають наступні компоненти.

1. **Корпус.** Корпус АПА виготовляється з матеріалів, стійких до впливу води і високого тиску. Завдяки спеціальній обтічній формі корпусу мінімізується опір води, що сприяє зменшенню енергоспоживання і підвищує стабільність у процесі руху. Деякі АПА мають модульний корпус, що дозволяє додавати чи змінювати обладнання для різних завдань.
2. **Система живлення.** Джерела живлення, такі як акумулятори або паливні елементи, забезпечують тривалу автономну роботу АПА. Енергоефективність є критичною для апаратів, які виконують місії на великих глибинах, де неможливо проводити регулярне підзарядження. Сучасні моделі апаратів оснащені системами, які дозволяють автоматично налаштовувати витрати енергії залежно від поставлених задач, щоб оптимізувати термін автономної роботи.
3. **Навігаційні та сенсорні системи.** Навігаційні системи, такі як GPS і акустичні сенсори, допомагають АПА точно визначати своє положення під водою. Крім того, гідролокатори, магнітометри, та спеціальні камери дозволяють отримувати інформацію про навколишнє середовище. Акустичні системи навігації є особливо важливими, оскільки GPS сигнал не працює під водою, і тому АПА покладаються на ехолокаційні технології для орієнтації в просторі. До того ж, гідроакустичні дані можна використовувати й для їх подальшої інтелектуальної обробки.
4. **Система керування та програмне забезпечення.** Для виконання завдань автономні підводні апарати обладнують інтегрованою інтелектуальною системою керування, яка обробляє інформацію з сенсорів і приймає рішення на основі попередньо закладеного алгоритмічного забезпечення. Програмне забезпечення дозволяє АПА адаптуватися до змін у навколишньому середовищі та обирати найбільш оптимальний маршрут для виконання місії. Деякі моделі можуть використовувати машинне

навчання для покращення навігації та розпізнавання перешкод та об'єктів інтересу.

5. **Комунікаційні системи.** Через особливості підводного середовища, автономні підводні апарати використовують різні типи зв'язку, включаючи акустичні сигнали для передачі даних. Швидкість та обсяг передачі інформації можуть бути обмежені під водою, тому акустична комунікація дозволяє апарату передавати базові дані в режимі реального часу.

Отже, автономні підводні апарати є складними технологічними системами, які дозволяють виконувати широкий спектр завдань у складних умовах. Їх автономність, маневреність, здатність працювати на великих глибинах та інтелектуальні можливості обробки даних роблять їх актуальними інструментами для досліджень, що сприяють розвитку науки і техніки, сприяють виконанню гуманітарних та військових операцій. Важливим елементом автономних апаратів є програмне забезпечення, що забезпечує їх функціонування та виконання поставлених задач. Цей факт підкреслює актуальність нашої роботи й необхідність створення комплексних інтелектуальних систем для автономних підводних апаратів на базі штучного інтелекту.

1.2 Огляд існуючих методів детектування об'єктів

Детектування об'єктів є однією з галузей штучного інтелекту, яка забезпечує вирішення низки прикладних задач. До них належить інтелектуальне відеоспостереження, автоматизований контроль дорожнього руху, автономне кермування, тощо. Під поняттям детектування зазвичай мають на увазі комплексну задачу, яка включає виділення ознак зображення, локалізація об'єкта на зображенні і його класифікація. Підходи до

детектування можна розділити на дві категорії: на основі класичного машинного навчання та методи на основі глибоких нейронних мереж.

1.2.1 Методи на основі машинного навчання

До появи сучасних нейронних мереж, детектування об'єктів виконувалось за допомогою класичних методів машинного навчання. Обмеженість наявних методологій виділення ознак з зображення і слабкі обчислювальні потужності тогочасних комп'ютерів призводили до того, що підходи до детектування часто базувались на ручному проектуванні ознак і подальшому використанні алгоритмів класифікації [1].

Одним з перших представників класичних методів детектування об'єктів є **алгоритм П. Віюлі й М. Джонса**, створений у 2001 році. Алгоритм використовувався для детектування обличчя на зображеннях і застосовував принцип плаваючого вікна — послідовної обробки частин зображення по всій його площі і виконанню пошуку цільового об'єкта у кожному з вікон [2]. Існували також вимоги до розміру вхідного зображення, положенню обличчя в кадрі та освітленості. Першим кроком обчислювались ознаки Хаара (рис. 1.3), які згодом порівнювались із набором значущих ознак, виділених алгоритмом Adaboost.

Авторами також була запропонована каскадна структура класифікаторів, за допомогою яких відсівалися області без об'єкту інтересу і аналіз продовжувався лише для тих вікон, які з вищою імовірністю містили потрібний об'єкт. Такий підхід дозволяв скоротити час обчислень за рахунок уникнення зайвих вікон.

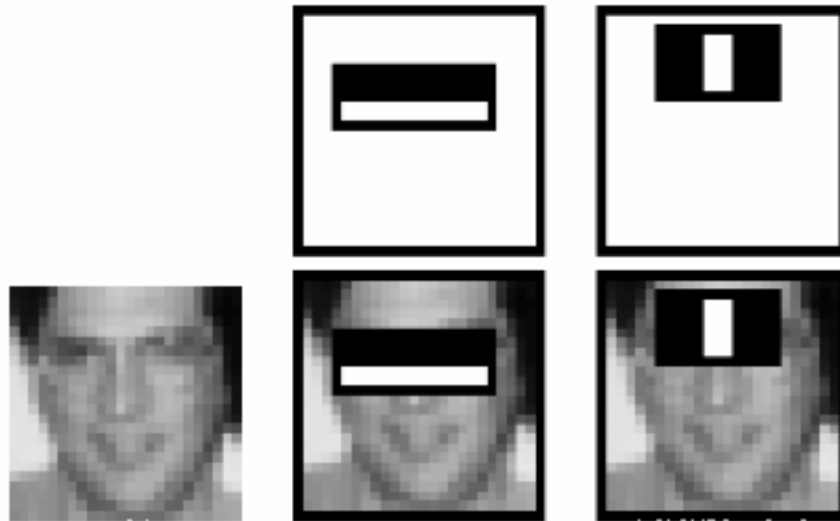


Рисунок 1.3 — Ознаки Хаара для розпізнавання обличчя

Іншим представником класичних алгоритмів розпізнавання об'єктів є метод гістограми напрямлених градієнтів (англ. — Histogram of Oriented Gradients, HOG) 2005 року [3]. Алгоритм вилучає інформацію про напрямки градієнтів у локальних областях зображення, які відображають зміни інтенсивності пікселів, що корелюють із краями, кутами та іншими деталями структури об'єкта. Зображення попередньо нормалізується та приводиться до фіксованого алгоритмом розміру. Наступним кроком обчислення градієнтів за допомогою ядра (наприклад, $[-1,0,1]$ і $[-1,0,1]^T$, або фільтра Собеля). Зображення розбивається на комірки — малі прямокутні області, для яких будуються гістограми градієнтів, та блоки — групи комірок, щоби забезпечити більшу стійкість до змін в яскравості. На основі цього відбувається побудова гістограм напрямків градієнтів, їх нормалізація та формування вектору ознак, який слугував вхідним значенням для алгоритмів класифікації.

Метод гістограми напрямлених градієнтів (рис. 1.4) успішно застосовувався для задачі визначення пішоходів на зображеннях дорожнього руху. Його перевагами порівняно з попередниками була інваріативність до змін у яскравості, проте цей метод також характеризувався високою

³ Джерело зображення — https://docs.opencv.org/4.x/d2/d99/tutorial_js_face_detection.html

обчислювальною складністю і недостатньою гнучкістю через врахування лише локального, а не просторового контексту.

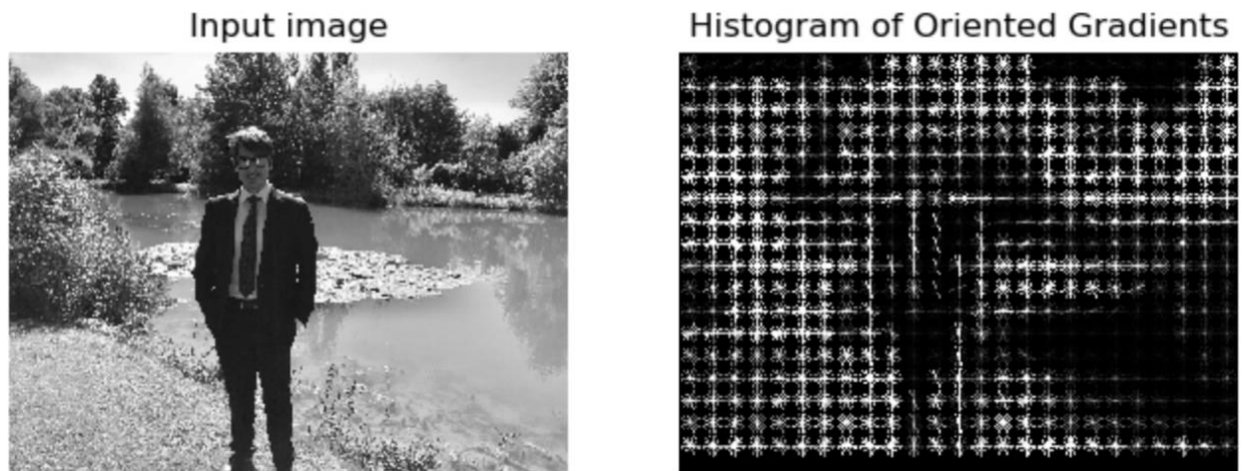


Рисунок 1.4 — Метод гістограми напрямлених градієнтів для детектування пішоходів

Вдосконаленням даного методу став алгоритм Deformable Part-based Model (скор. — DPM), що вийшов у 2008 році [4]. Автори алгоритму П. Фельценцвальб і Р. Гіршік підійшли до задачі детектування використовуючи парадигму «розподіляй і володарюй» (англ. — divide and conquer), яка передбачає розбиття однієї складної задачі на ряд окремих підзадач, які можна простіше вирішити окремо.

Алгоритм DPM описує об'єкт через головний фільтр (англ. — root filter) та фільтри частин (англ. — part filters). За допомогою навчання зі слабким керуванням конфігурації фільтрів частин вивчаються автоматично як латентні змінні. Зображення сканується за принципом плаваючого вікна, і для кожного з його положень обчислюється відповідність головній моделі та виконується пошук частин об'єкта на основі фільтрів частин. Порівняно з минулими реалізаціями алгоритмів детектування, DPM дозволяв враховувати деформації об'єкта на зображенні — зміну поз, часткове перекриття, тощо.

⁴ Джерело зображення — <https://github.com/SamPlvs/Object-detection-via-HOG-SVM>

Також частина понять (напр., задача регресії обмежувальних рамок), якими оперували автори методу, слугували основою для сучасніших алгоритмів на базі нейронних мереж.

1.2.2 Методи на основі нейронних мереж

Основною проблемою алгоритмів детектування об'єктів, заснованих на класичних методах машинного навчання, були недосконалі методи вилучення ознак. З розвитком згорткових мереж (англ. — convolutional neural networks, CNN) з'явилися нові можливості по формуванню інформативних ознак з вхідного зображення. Інтеграція глибокого навчання в методології детектування об'єктів значно пришвидшила їх розвиток та вивела ефективність на новий рівень. Наразі, методи детектування об'єктів на базі нейронних мереж можна поділити на дві категорії:

- методи на базі згорткових нейронних мереж;
- методи на основі архітектури Transformer.

В свою чергу, методи на базі згортки можна поділити на двокрокові алгоритми, в яких виділення області інтересу і пошук об'єкту в них є окремими підзадачами, та однокрокові алгоритми, які водночас і локалізують об'єкт, і класифікують його [5].

Першим представником двокрокових алгоритмів детектування є R-CNN [6]. Алгоритм починає роботу з вилучення локальних зон зображення, які слугуватимуть кандидатами рамки, що міститиме об'єкт. Кожна з прогнозованих областей обробляється згортковою нейронною мережею для вилучення ознак, формування їх вектору і подальшої класифікації окремою нейронною мережею.

За рахунок поліпшеної репрезентації ознак, R-CNN значно випереджала класичні рішення за точністю, однак відрізнялась значним недоліком —

кількість визначених областей-кандидатів для одного зображення зазвичай вимірялася тисячами, і кожне з цих вікон оброблялось окремо, що призводило до значних затримок при обробці зображень.

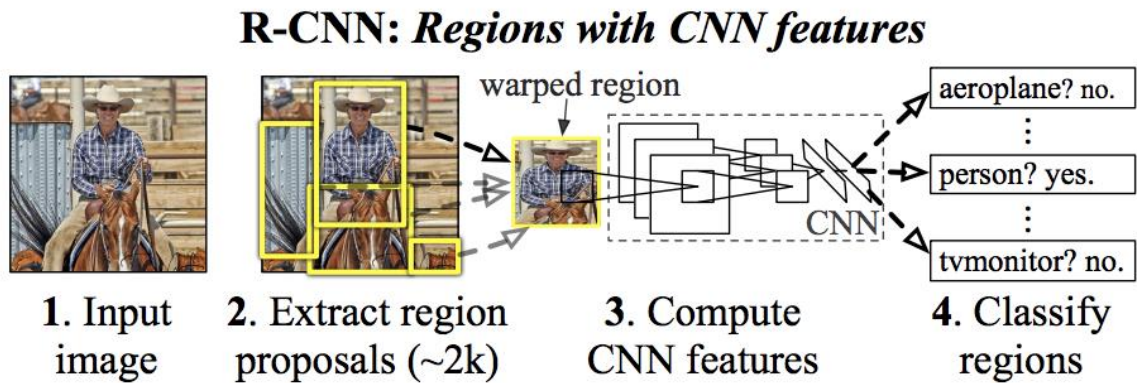


Рисунок 1.5 — Архітектура мережі R-CNN

Проблема довгої тривалості обробки зображень була вирішена в мережі SPPNet [7].

Замість підходом з визначенням областей, які ймовірно містять об'єкт, в SPPNet згортова нейронна мережа вилучала ознаки зображення, а шар Spatial Pyramid Pooling (скор. — SPP) відповідав за генерацію вихідних векторів фіксованої довжини, які подавались на вхід повнозв'язних шарів мережі, що відповідали за класифікацію (рис. 1.6).

В результаті такого підходу, швидкість детектування об'єктів виросла в понад 20 разів, порівняно з R-CNN. Однак алгоритм SPPNet був наділений також і значним недоліком: в такій архітектурній конфігурації мережі навчанню підлягали лише повнозв'язні шари, а не згорткові, що призводило до меншої точності детектування, й не мало простого вирішення через особливості архітектури.

⁵ Джерело зображення — <https://blog.roboflow.com/what-is-r-cnn/>

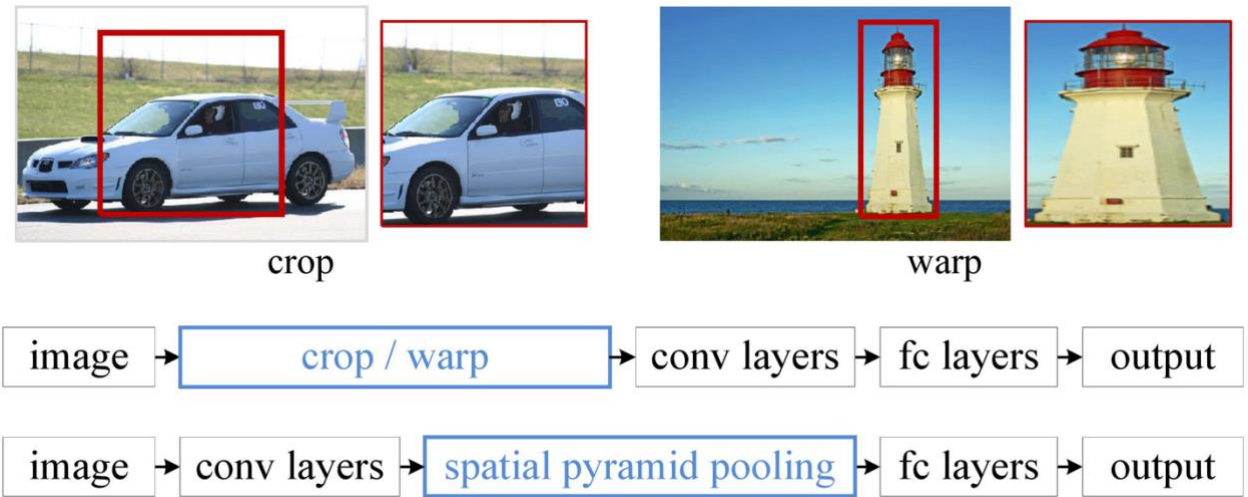


Рисунок 1.6 — Принцип роботи мережі SPPNet

Метод Fast R-CNN, який вийшов на рік пізніше, вирішив дану проблему і дозволив одночасно навчати обидві мережі [8]. Вилучення ознак було покладене на згорткову мережу, а наступним кроком вектори ознак і обмежувальні рамки подавались на шар пулінгу для отримання векторів фіксованої довжини, які в свою чергу слугували вхідними даними для шарів регресії та класифікації.

Заключною в цій серії стала мережа Faster R-CNN [9], в якій вперше відмовились від класичного підходу до генерації областей-кандидатів і доручили цю задачу окремому модулю нейронної мережі під назвою Region Proposal Network (скор. — RPN), що значно скоротило обчислювальні витрати за рахунок одночасної обробки всього зображення, а не окремих його областей. Швидкість та точність Faster R-CNN випереджали існуючі на той час рішення, і були покращені пізніше завдяки додаванню в архітектуру мережі піраміди ознак (англ. — Feature Pyramid Network, FPN) [10]. Впровадження даної структури дозволило агрегувати ознаки з різних шарів глибокої нейронної мережі, збагаючи глобальний контекст за рахунок використання різних розмірів карт ознак (рис. 1.7).

⁶ Джерело зображення — <https://paperswithcode.com/method/spp-net>

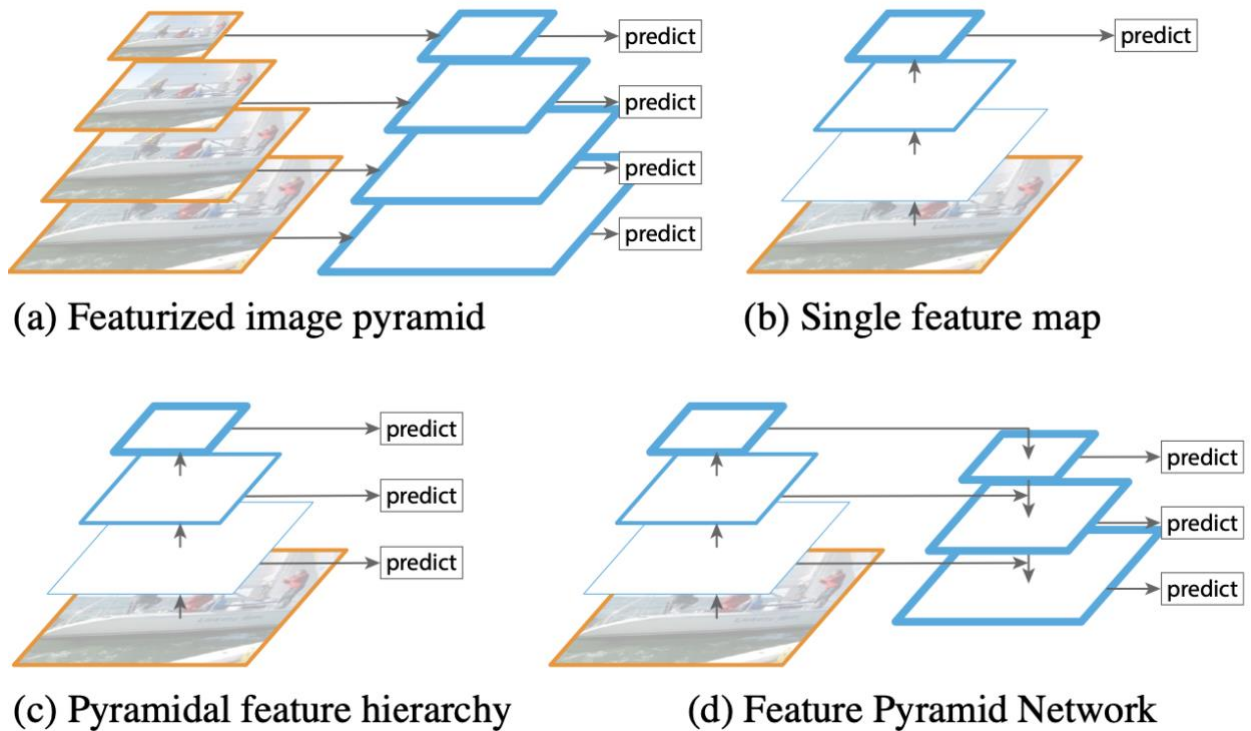


Рисунок 1.7 — Архітектура Feature Pyramid Network

Паралельно з цим, з 2015 року почали розвиватися однокрокові алгоритми детектування об'єктів, в яких задача регресії і класифікації розв'язуються за один прохід по зображенню. Найпершим і найбільш активним у розвитку однокроковим методом детектування є You Only Look Once (скор. — YOLO), розроблений Дж. Редмоном у 2015 році [11]. YOLO працює на базі поділу зображення на блоки і одночасного передбачення обмежувальних рамок об'єктів та їх класів.

З часу виходу алгоритм зазнав багато значних змін та покращень, які були спрямовані як на покращення в точності й швидкості, так і на пристосування до різних цільових задач. Наприклад, окрім задачі виявлення об'єктів, YOLO також здатен розв'язувати задачі сегментації та детектування пози.

Детально алгоритми сімейства YOLO будуть розглянуті в розділі 2.

⁷ Джерело зображення — <https://arxiv.org/pdf/1612.03144v2>

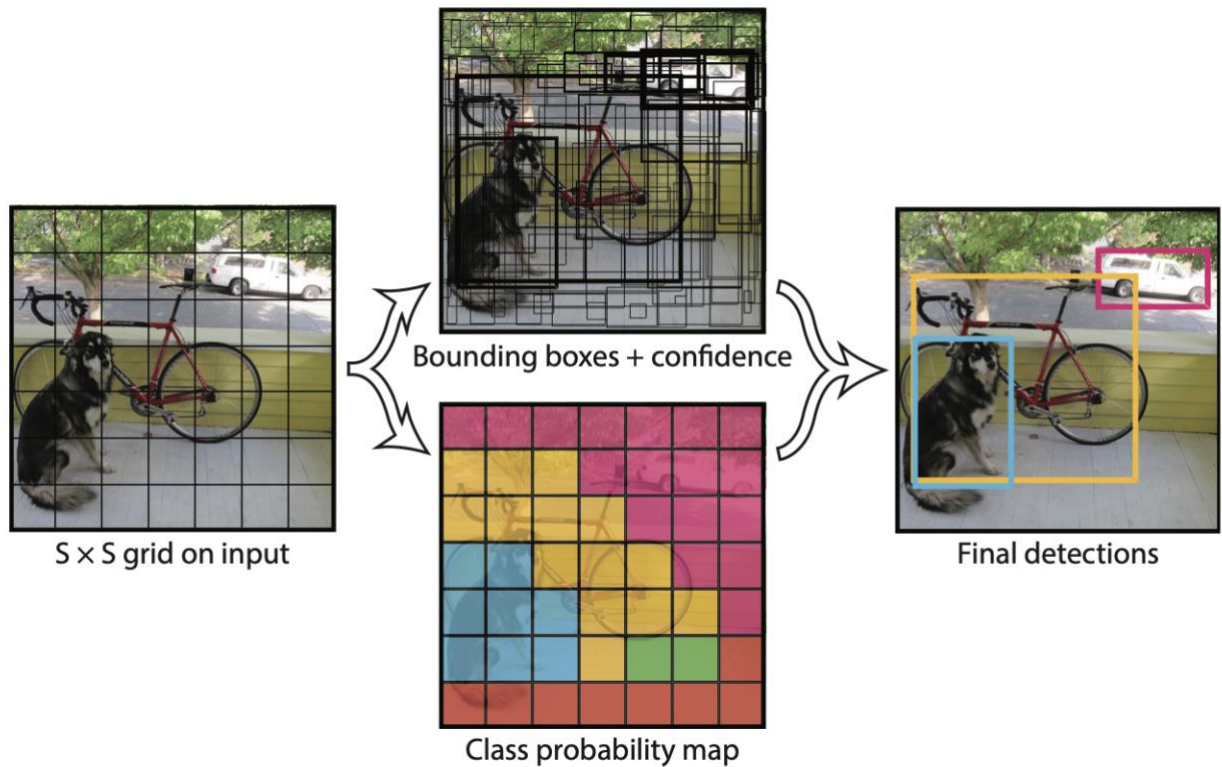


Рисунок 1.8 — Принципи роботи першої версії YOLO, запропоновані Дж.

Редмоном: поділ зображення на блоки, обрахування ймовірності
обмежувальних рамок і карти класів

Альтернативним методом, який теж виконує детектування в один прохід, є Single Shot Detector (SSD) [12]. Метод базується на використанні так званих «якорів» з різними співвідношеннями сторін і розмірами для прогнозування об'єктів різних масштабів. Для вилучення ознак використовуються глибокі згорткові мережі, такі як VGG або ResNet, а додаткові згорткові шари використовуються для побудови багаторівневої архітектури для розпізнавання об'єктів, різних за масштабом — кожен додатковий шар має меншу просторову роздільну здатність, але більші рецептивні поля, що дозволяє детектувати більші об'єкти.

⁸ Джерело зображення — <https://arxiv.org/pdf/1506.02640>

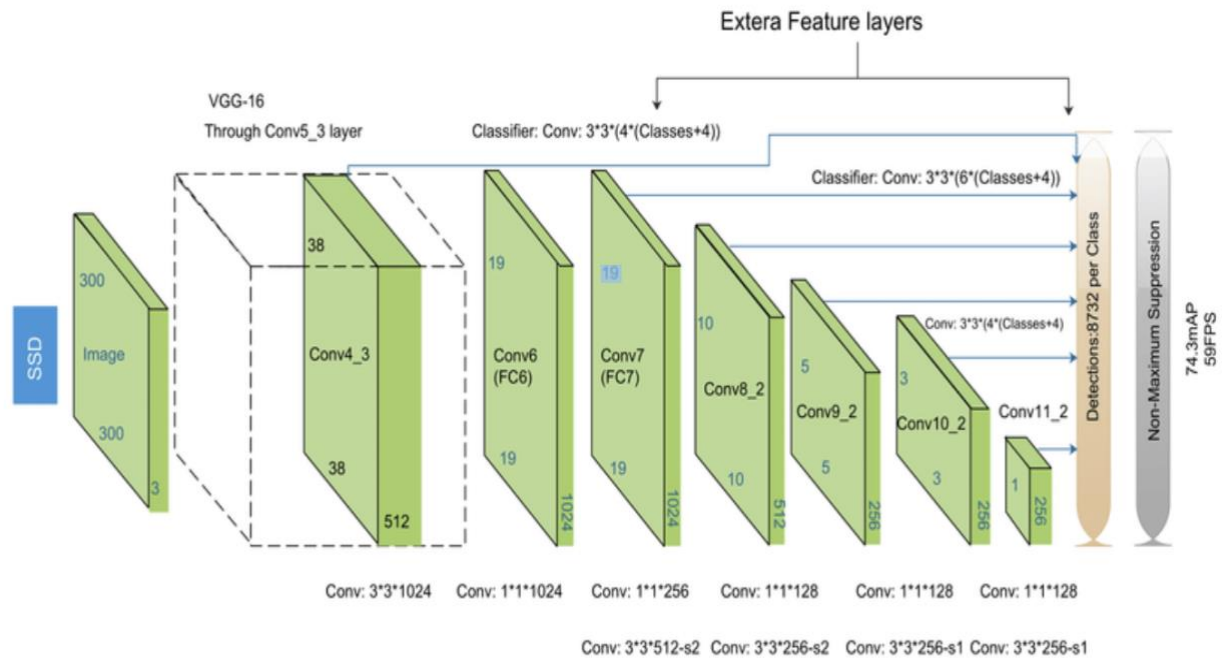


Рисунок 1.9 — Будова Single Shot Detector (SSD)

Окрім архітектурних рішень, побудованих виключно на базі згортки, активно розвиваються також і моделі детектування об'єктів на базі архітектури Transformer [13]. Вони дозволяють виключити зайві обчислювальні кроки, викликані використанням згорткових нейронних мереж, наприклад, відсів хибних рамок алгоритмами Non-Maximum Suppression (скор. — NMS).

Одним з представників таких методів детектування є Detection Transformer (скор. — DETR) [14]. DETR все ще використовує згорткову мережу для вилучення ознак, які отримують позиційне кодування і потрапляють у мережу формату енкодер-декодер. Модель детектування навчається характеристикам зображення і передає вихідні вектори на нейронну мережу для класифікації. Такий підхід є простішим, ніж ті, що базуються на згортці, однак для ефективного навчання DETR потрібно набагато більше ітерацій та більша навчальна вибірка.

⁹ Джерело зображення — https://www.researchgate.net/figure/Single-Shot-Multi-Box-Detector-SSD-architecture-47_fig4_340654462

1.3 Проблематика створення інтелектуальних систем для автономних підводних апаратів

Сучасні системи розпізнавання об'єктів, розроблені на основі глибоких нейронних мереж, демонструють значні успіхи у звичайних середовищах [15]. Проте їх ефективність значно знижується в умовах підводного простору. Це обумовлено рядом факторів, які накладають обмеження на функціональність таких систем та ускладнюють процес створення інтелектуальних систем для автономних підводних апаратів (рис. 1.10).



Рисунок 1.10 — Приклад підводного знімка з проблемою неоднорідного фону, що ускладнює детектування об'єктів.

Підводне середовище створює унікальні труднощі для розпізнавання об'єктів. По-перше, розпізнавання на зображеннях, отриманих з сенсорів та

¹⁰ Джерело зображення — <https://universe.roboflow.com/srmist-vx65l/utdac2020-kkoqh>

камер автономних роботів, часто ускладнюється наявністю дрібних цілей, що можуть бути складними для виявлення. Малий розмір об'єктів зменшує їхню розрізняюваність на фоні, що значно ускладнює задачу детектування.

По-друге, об'єкти часто накладаються один на одного, що ускладнює навчання нейронної мережі та розрізнення щільно розташованих об'єктів (рис. 1.11). Оскільки підводне середовище не має однорідного фону, об'єкти на зображеннях можуть зливатися з навколишніми елементами, такими як мул, каміння чи коралові рифи, що значно ускладнює детектування [16].

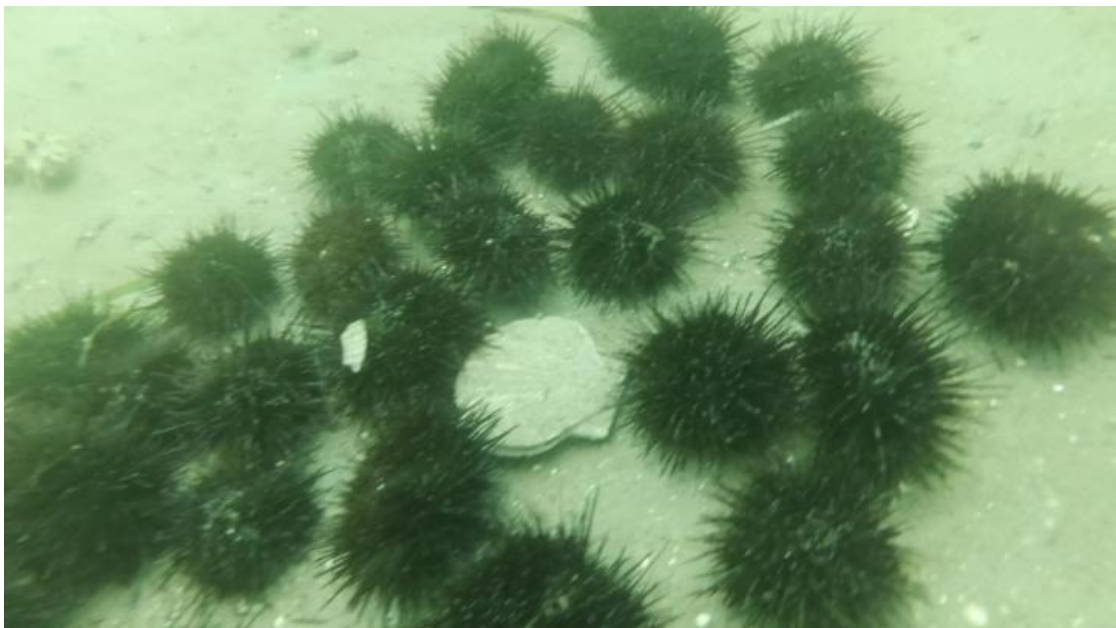


Рисунок 1. 11 — Приклад щільного розташування об'єктів в підводному середовищі. Щільне розташування об'єктів під водою часто притаманне вибіркам, що стосуються біорізноманіття.

Часткове перекриття об'єктів (рис. 1.12) також є перешкодою для ефективного детектування. Наприклад, цільові об'єкти під водою можуть бути частково покриті мулом, водоростями, камінням або приховані за підводними спорудами, що має враховуватись методами виявлення об'єктів. Також слід зауважити, що об'єкти на різних зображеннях можуть мати різні масштаби,

¹¹ Джерело зображення — <https://universe.roboflow.com/srmist-vx65l/utdac2020-kkoqh>

залежно від їхньої відстані до камери та глибини, на якій вони розташовані. Така варіативність у розмірах, формах об'єктів та типах перешкод додає до складності завдання, адже моделі повинні бути здатні до масштабованого розпізнавання, зберігаючи високу точність.

Іншою важливою проблемою є недостатня різноманітність вибірок підводних знімків, які використовуються для навчання моделей розпізнавання об'єктів для автономних апаратів. Більшість доступних вибірок містять обмежений набір зображень, що не відображає усіх можливих сценаріїв, з якими система може зіткнутися в реальних умовах. Наприклад, однаковий об'єкт може виглядати по-різному залежно від глибини, ступеня освітленості та кута зйомки. Це викликає проблему недостатньої здатності натренованих моделей до узагальнення, коли модель, навчена на обмеженому наборі зображень, може неефективно працювати з новими, раніше невідомими даними.



Рисунок 1.12 — Приклад проблеми з частковою прихованістю об'єкта за іншим

¹² Джерело зображення — <https://universe.roboflow.com/srmist-vx65l/utdac2020-kkoqh>

Ще одна особливість підводного середовища — це неструктурованість форми об'єктів. Наприклад, водорості можуть мати довільну форму та змінювати її під впливом течії, що ускладнює їх розпізнавання традиційними методами, орієнтованими на розпізнавання чітких контурів і форм. Нерівномірне освітлення під водою також може призводити до візуальних спотворень, які здатні погіршувати здатність моделей розпізнавати об'єкти.

Для подолання цієї проблеми необхідні більш комплексні та різноманітні підводні вибірки, що охоплюють широкий спектр умов, різноманітних об'єктів та середовищ. Створення таких наборів даних потребує значних ресурсів і часто є технічно складним та фінансово обтяжливим процесом.

Однак ще однією значущою проблемою є обмеженість обчислювальних ресурсів, доступних на автономних пристроях. Оскільки апарати часто працюють у віддалених зонах і на великих глибинах, передача великих обсягів даних на наземні центри для обробки є складною через обмежену пропускну здатність та нестабільність каналів зв'язку. Для обробки даних в режимі реального часу потрібно здійснювати розпізнавання об'єктів безпосередньо на апараті, що вимагає значних обчислювальних ресурсів і ефективного управління ними.

Автономні підводні апарати мають балансувати ресурси між завданнями розпізнавання об'єктів та іншими функціями, такими як навігація, уникнення перешкод та управління пристроями. Це створює додаткове навантаження на апаратну частину та підвищує вимоги до оптимізації системи. Водночас, для багатьох сценаріїв, особливо у військових або рятувальних операціях, критичним є швидке реагування і низька затримка обробки даних, що додає ще одну перешкоду при створенні інтелектуальних рішень для АПА.

Зважаючи на зазначені виклики, розробка інтелектуальної системи для автономних підводних апаратів, яка здатна працювати в складних підводних умовах, є комплексною та важливою задачею.

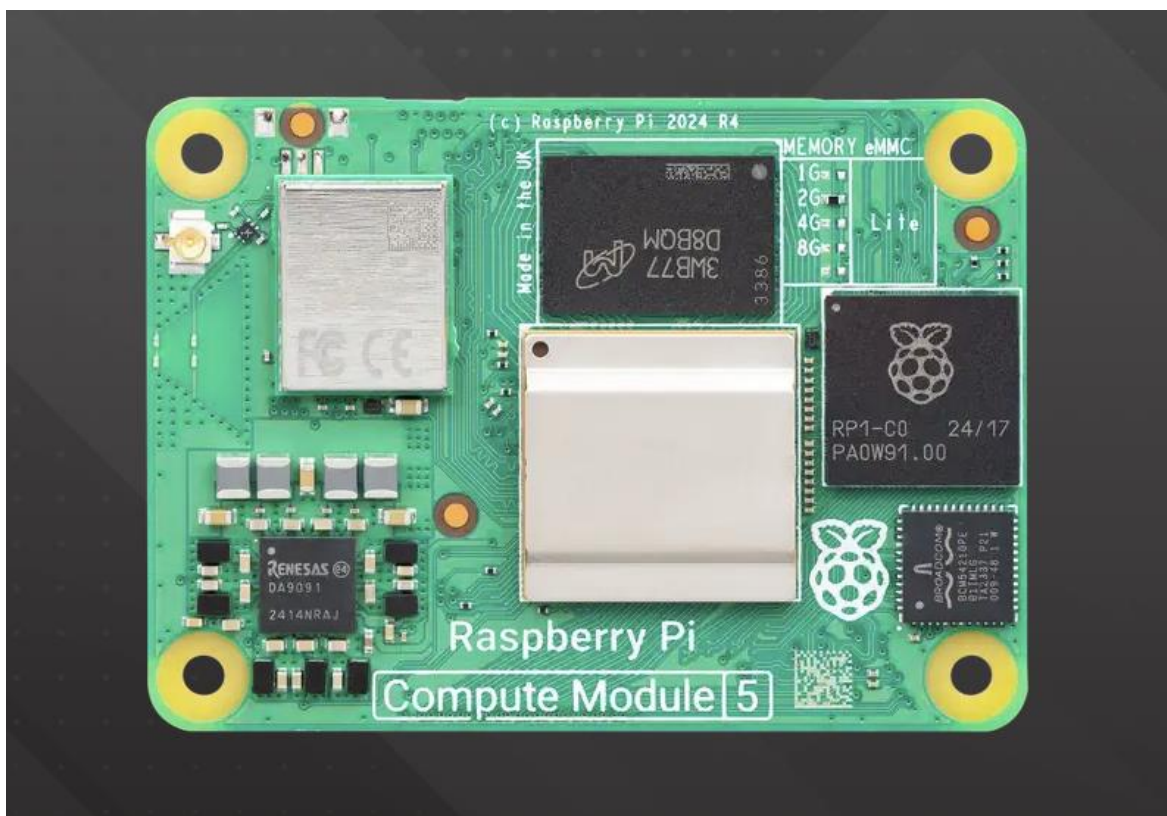


Рисунок 1.13 — Raspberry Pi — один з прикладів модулів, на який встановлюється інтелектуальна система обробки зображень на автономному апараті

Створення такої системи допоможе збільшити спектр можливостей підводних роботів в умовах обмежених обчислювальних ресурсів, недостатньої видимості та дозволить точно визначати об'єкти різного масштабу та форми. Наявність такого алгоритмічного та програмного забезпечення значно підвищить ефективність роботи автономних підводних апаратів у місіях різного рівня складності. Окрім того, нейронна мережа повинна бути обчислювально ефективною для можливості розгортання на вбудованому апаратному забезпеченні автономних підводних апаратів у якості зовнішнього модуля. Така інтеграція інтелектуальної системи і апаратного забезпечення дозволила б автономним підводним апаратам працювати більш ефективно в складних підводних умовах.

¹³ Джерело зображення — <https://www.raspberrypi.com/products/compute-module-5/?variant=cm5-104032>

1.4 Математична постановка задачі детектування об'єктів

Задачу структурно-параметричного синтезу детектора об'єктів для роботи в підводному середовищі будемо розглядати як задачу пошуку оптимальної топології нейронної мережі та зменшення її розмірів шляхом дистиляції знань від попередньо навченої нейронної мережі більшого розміру. Задача структурно-параметричного синтезу також включає в себе вибір початкових параметрів ініціалізації мережі, гіперпараметрів та методу оптимізації [17].

Вирішення даної задачі передбачає побудову нейронної мережі M , що приймає на вхід зразки з навчальної вибірки $(X_1, y_1), (X_2, y_2), \dots, (X_i, y_i)$, де X_i позначає i -ту RGB матрицю зображення розмірністю $n \times n \times 3$, а y_i позначає вектор істинних координат обмежувальної рамки, яка позначає розташування об'єкта на даній матриці зображення та мітки класів для кожного об'єкта на даному зображенні. Модель нейронної мережі M параметризована вагами θ , здійснює наступне перетворення:

$$f_{\theta}: X_i \mapsto \{(\hat{b}_{ij}, \hat{c}_{ij})\}_{j=1}^k,$$

де $\hat{b}_{ij}$ це передбачена обмежувальна рамка для об'єкту j на зображенні X_i , а $\hat{c}_{ij}$ це передбачені ймовірності класів для кожного з об'єктів на даному зображенні. Задача полягає у мінімізації розбіжності між передбаченими $\hat{b}_{ij}, \hat{c}_{ij}$ і істинними b_{ij}, c_{ij} координатами рамок та міток класів передбачених об'єктів, відповідно.

Загальна функція витрат нейронної мереж детектування об'єктів визначатиметься як сума функцій витрат регресії та класифікації, помножених на відповідні вагові коефіцієнти:

$$\mathcal{L} = \lambda_{reg}\mathcal{L}_{reg} + \lambda_{cl}\mathcal{L}_{cl},$$

де λ_{clou} і λ_{vfl} це гіперпараметри, що зважують важливість кожної з функцій у рамках загальної функції витрат.

Функції витрат нейронних мереж детектування навчаються алгоритмами на базі зворотного поширення градієнту. Загальне оновлення вагів на кожній ітерації навчання можна сформулювати як:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \mathcal{L},$$

де η це коефіцієнт швидкості навчання, що керує розміром кроку в протилежну від градієнта сторону, а $\nabla_{\theta} \mathcal{L}$ це градієнт функції витрат відповідно до параметрів моделі.

Висновки до розділу 1

У результаті проведеного аналізу ми встановили, що автономні підводні апарати мають великий потенціал у різних сферах, таких як наукові дослідження, моніторинг екологічного стану та військові операції. Однак, створення інтелектуальних систем для таких апаратів є складним завданням через особливості підводного середовища: низька видимість, складний фон, малі розміри об'єктів, часткове перекриття та їх різні масштаби. Сучасні методи детектування підводних об'єктів мають обмеження — від складності мережових архітектур до недостатньої здатності до узагальнення.

Розглянуто наявні методи та алгоритми виявлення об'єктів. Наведено їх класифікацію та історію розвитку, починаючи від класичних методів на базі машинного навчання до сучасних методологій, що засновані на принципах глибокого навчання. Розглянуті переваги та недоліки різних методів. Методи

детектування об'єктів сімейства YOLO обрано за основу подальшого дослідження та взято до детального аналізу їх історії, топології та алгоритмічної основи.

Для вирішення завдання створення інтелектуальної системи детектування об'єктів для автономних підводних апаратів було наведено математичну постановку задачі структурно-параметричного синтезу нейронної мережі детектування об'єктів.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ НЕЙРОННОЇ МЕРЕЖІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ

2.1 Попередня обробка зображень

Попередня обробка зображень є критично важливим етапом у інтелектуальних системах детектування об'єктів. Даний процес спрямований на підвищення якості зображень у ряді характеристик, зокрема покращення контрастності зображення (рис. 2.1), усунення артефактів, підвищення роздільної здатності, зменшення рівню шумів та усунення небажаних ефектів, таких як розмиття (англ. — blur), затуманеність (англ. — fog) та викривлення кольору (англ. — color cast).

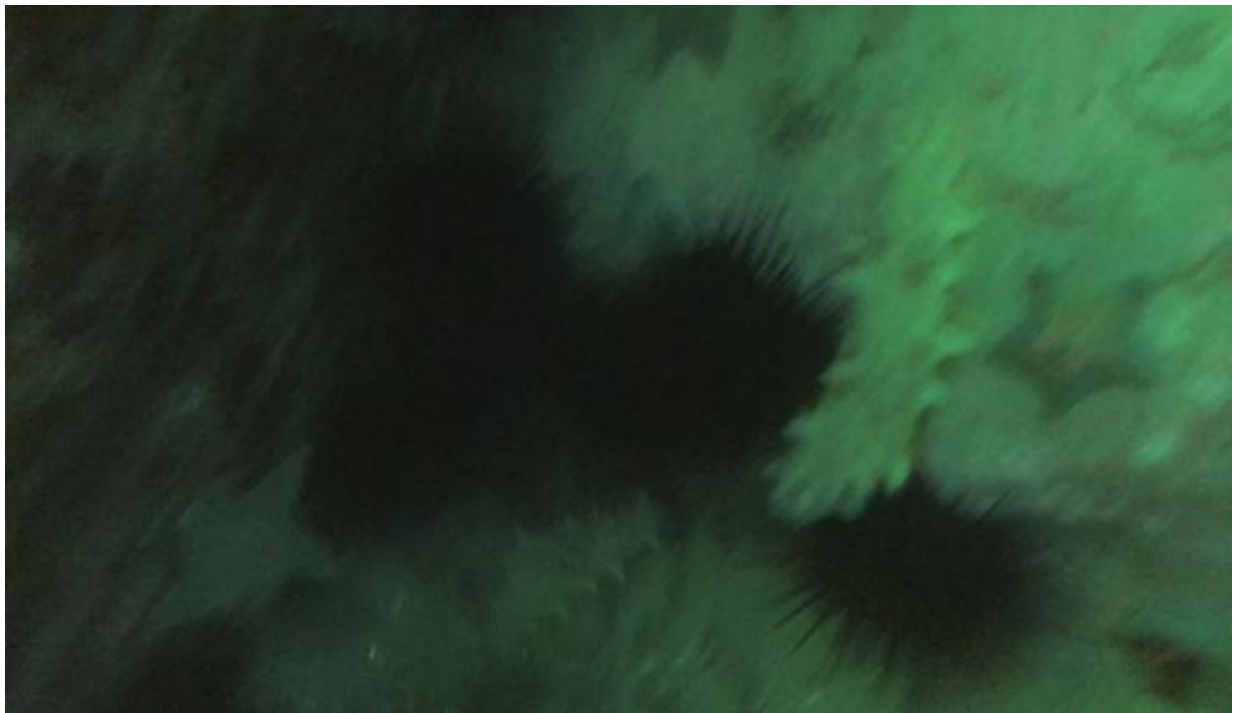


Рисунок 2.1 — Приклад зображення, знятого у підводному середовищі. Даний знімок має проблеми з контрастністю, а об'єкти інтересу знаходяться у затемненій області.

¹ Джерело зображення — <https://universe.roboflow.com/srmist-vx65l/utdac2020-kkoqh>

Далі зображення з покращеними характеристиками використовується для більш ефективного навчання моделей машинного навчання та нейронних мереж. Для знімків, зроблених у підводному середовищі, етап попередньої обробки має критичне значення, через низку особливостей таких зображень, описаних у розділі 1.

Розглянемо основні категорії методів попередньої обробки, що використовуються для підводних зображень, та їх алгоритмічну основу.

Одним з базових методів підвищення контрастності зображень є гістограмна еквалізація (англ. — histogram equalization, HE) [18]. Даний метод ґрунтується на вирівнюванні розподілу інтенсивності пікселів зображення (рис. 2.2). В результаті роботи алгоритму, гістограма не має пікових значень і є рівномірною, а отже, покращується деталізація на зображеннях з проблемами контрастності.

Математично, метод гістограмної еквалізації полягає в наступному. Нехай $I(x, y)$ позначає інтенсивність пікселя у точці (x, y) , що належить до діапазону $[0, L - 1]$, де L це кількість рівнів яскравості. Тоді гістограму зображення можна визначити як функцію частот $p_r(r_k)$, де r_k позначає рівень яскравості, а $p_r(r_k)$ — ймовірність його появи:

$$p_r(r_k) = \frac{n_k}{N}, \sum_{k=0}^{L-1} p_r(r_k) = 1,$$

де n_k це кількість пікселів з яскравістю r_k , а N — загальна кількість пікселів у зображенні. Кумулятивна функція розподілу визначається як:

$$CDF(r_k) = \sum_{i=0}^k p_r(r_i),$$

а нова інтенсивність кожного пікселя визначатиметься ерез лінійне розтягнення даної функції на діапазон $[0, L - 1]$ як:

$$s_k = \lfloor (L - 1) \cdot CDF(r_k) \rfloor.$$

В результаті отримуємо зображення з рівномірною гистограмою. Проте недоліком даного алгоритму є те, що при обробці зображення не враховується різниця в яскравості між різними ділянками на зображенні.

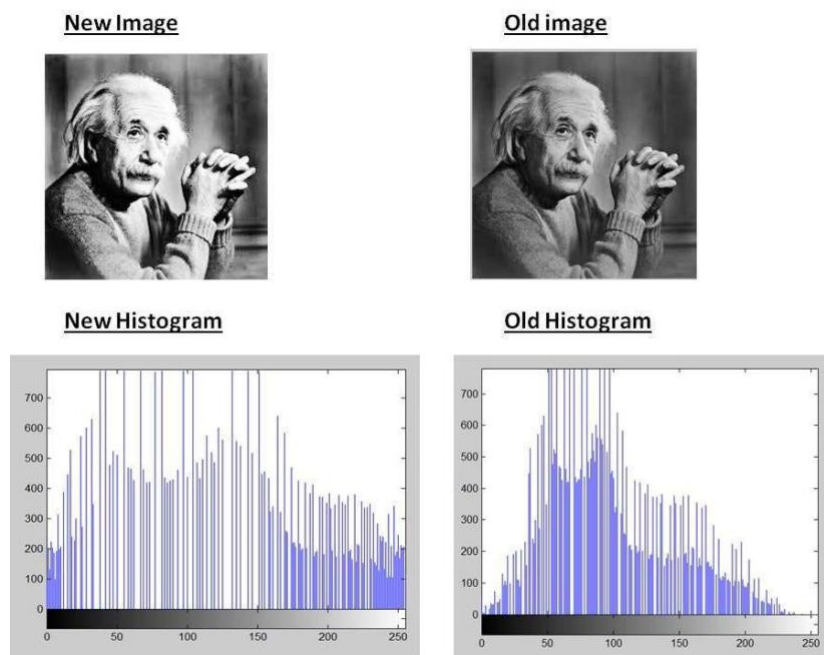


Рисунок 2.2 — Результат роботи алгоритму гістограмної еквалізації (НЕ)

Даний недолік вирішено у алгоритмі адаптивної гістограмної еквалізації (англ. adaptive histogram equalization, АНЕ) [19]. Ідея даного методу полягає у розбивці зображення на окремі локальні області і їх подальшій окремій обробці, що забезпечує вищу якість при роботі з зображеннями з нерівномірним освітленням. Для цього окрім інтенсивності пікселів задається розмір вікна W і для кожного вікна $W_{i,j}$ виконується розрахунок локальної

² Джерело зображення — <https://medium.com/@rajilini/histogram-equalization-all-you-want-to-know-13b0884c5296>

гістограми $p_r^{i,j}(r_k)$ у межах вікна та розрахунок кумулятивної функції розподілу $CDF_{i,j}(r_k)$ як:

$$CDF_{i,j}(r_k) = \sum_{m=0}^k p_r^{i,j}(r_m),$$

на основі чого інтенсивності пікселів обраховується як:

$$s(x, y) = \lfloor (L - 1) \cdot CDF_{i,j}(I(x, y)) \rfloor.$$

Проте такий підхід призводить до підсилення рівня шумів у локальних областях знімка зі слабким сигналом. Введення додаткового обмеження на максимальну висоту гістограми для кожного локального вікна дозволяє уникнути надмірної ампліфікації шуму. Дана модифікація методу називається контрастно-обмеженою адаптивною гістограмною еквалізацією (англ. contrast-limited adaptive histogram equalization, CLAHE) (рис. 2.3) [20].



Рисунок 2.3 — Результат роботи алгоритму контраст-обмеженої гістограмної еквалізації (CLAHE)

³ Джерело зображення — <https://www.nomidl.com/computer-vision/mastering-image-contrast/>

Іншим алгоритмом попередньої обробки, що широко використовується для підводних знімків, є багатомасштабний ретинекс з відновленням кольору (англ. — Multiscale Retinex with Color Restoration, MSRCR). На відміну від сімейства алгоритмів гістограмної еквалізації, MSRCR працює не лише з інтенсивністю окремих пікселів зображення, а й з кольоровим балансом, базуючись на ретинекс-теорії, що імітує сприйняття кольору і яскравості людським оком. Даний метод виконує дві трансформації зображення. Першим кроком є багатомасштабна ретинекс корекція. Для кожного масштабу виконується підрахунок:

$$R_c^n(x, y) = \log(I_c(x, y)) - \log(F_n(x, y) * I_c(x, y)),$$

де $I_c(x, y)$ це інтенсивність пікселя для кольорового каналу c , $F_n(x, y)$ функція Гаусівського розмиття для масштабу n , $*$ позначає операцію згортки. Багатомасштабна корекція відбувається як:

$$R_c(x, y) = \sum_{n=1}^N w_n \cdot R_c^n(x, y),$$

де w_n позначає вагові коефіцієнти. Наступним кроком відбувається кольорокорекція:

$$C_c(x, y) = \alpha \cdot \left(\frac{I_c(x, y)}{\sum_c I_c(x, y)} + \beta \right) \cdot R_c(x, y),$$

де α і β виступають параметрами, що відповідають за відновлення кольору та контрасту. Значним недоліком даного алгоритму є висока обчислювальна складність і тривалий час виконання, що в контексті підводних знімків унеможливорює роботу в реальному часі.

Для попередньої обробки зображень з ефектами мутності або затуманення використовують метод Dark Channel Prior (DCP) [21]. Цей підхід засновано на припущенні, що принаймні один канал у локальній області в незатуманеному зображенні має низькі інтенсивності (так званий темний канал). На його основі обчислюється ступінь замутинення, спричиненого середовищем, і реконструюється зображення без небажаних ефектів. Інтенсивність зображення $I(x, y)$ для пікселя в точці (x, y) описується як

$$I(x, y) = J(x, y)t(x, y) + A(1 - t(x, y)),$$

де $J(x, y)$ інтенсивність пікселів зображення без замутинення, $t(x, y)$ карта пропускання світла (англ. - transmission map), A глобальне освітлення середовища, $I(x, y)$ пікселі зображення.

Темний канал обчислюється як:

$$J^{dark}(x, y) = \min_{c \in \{R, G, B\}} \left(\min_{(x', y') \in \Omega(x, y)} J_c(x', y') \right),$$

де $\Omega(x, y)$ це локальна область навколо пікселя (x, y) , $J_c(x', y')$ це значення інтенсивності в кольоровому каналі c у локальній області. Спостерігається, що значення J^{dark} у замутинених регіонах зображення буде наближатись до нуля. Наступним кроком обчислюється карта пропускання світла:

$$t(x, y) = 1 - \omega \cdot \min_c \left(\min_{(x', y') \in \Omega(x, y)} \frac{I_c(x', y')}{A_c} \right),$$

де ω це параметр, що контролює усунення ефекту замутинення.

Фінальним кроком алгоритму є відновлення зображення на основі обрахованої карти:

$$J(x, y) = \frac{I(x, y) - A}{\max(t(x, y), t_0)} + A,$$

де t_0 це мінімальне значення інтенсивності з карти пропускання світла t . Результатом роботи алгоритму є зображення з усуненими ефектами затуманення та мутності (рис. 2.4).



Рисунок 2.4 — Результат роботи алгоритму Dark Channel Prior (DCP)

Окрім методів, зазначених вище, існують також підходи до попередньої обробки зображень на основі генеративно-змагальних нейронних мереж. Наприклад, для усунення ефекту розмиття (рис. 2.5), спричиненого рухом автономного апарата при зйомці в підводному середовищі використовують мережу Deblur GAN [22].

⁴ Джерело зображення — <http://mmlab.ie.cuhk.edu.hk/archive/2011/Haze.pdf>



Рисунок 2.5 — Результат роботи мережі Deblur GAN

Deblur GAN складається з двох частин: генератора і дискримінатора. Генератор G створює відновлене зображення, позбавлене небажаних ефектів розмиття, в той час як дискримінатор D оцінює ймовірність того, що зображення було створено штучно генератором. Навчання мережі виконується шляхом оптимізації наступної функції витрат:

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda \cdot \mathcal{L}_{perc},$$

де $\mathcal{L}_{GAN}$ це адверсаріальна (змагальна) втрата, $\mathcal{L}_{perc}$ це перцептивна (контекстуальна втрата). В якості змагальної втрати Deblur GAN використовує Wasserstein GAN зі штрафом градієнта (англ. – gradient penalty, GP):

$$\mathcal{L}_{GAN} = \mathbb{E}_{I_S}[D_{\theta_D}(I_S)] - \mathbb{E}_{I_B}[D_{\theta_D}(G_{\theta_G}(I_B))],$$

⁵ Джерело зображення — <https://medium.com/sicara/keras-generative-adversarial-networks-image-deblurring-45e3ab6977b5>

де D_{θ_D} – мережа-дискримінатор, параметризована вагами θ_D . В якості персептивної втрати в Deblur GAN розраховується різниця між ознаками, вилученими згортковою нейронною мережею VGG-19:

$$\mathcal{L}_{perc} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I_S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I_B))_{x,y})^2,$$

де $\phi_{i,j}$ це ознаки, вилучені на j –тому шарі i –тої згортки, а $W_{i,j}H_{i,j}$ описують розміри ознак. На виході генеративно-змагальної мережі отримуємо зображення з відновленими текстурами і деталями порівняно з вхідним зображенням з ефектом розмиття. Однак час обробки зображення даним методом підходить для використання лише на статичних зображеннях, а не в режимі реального часу.

2.2 Методи YOLO для задач детектування

2.2.1 Загальна характеристика методів YOLO

Алгоритм You Only Look Once (англ., досл. «ти дивишся лише раз», скор. YOLO) було розроблено американським вченим Джозефом Редмоном у 2015 році як уніфікований метод детектування об'єктів за допомогою нейронної мережі [11].

Принциповою новизною методу стало те, що для задачі локалізації і класифікації об'єктів була вперше застосована одна нейронна мережа, звідки й назва (рис. 2.6). До винайдення YOLO задача детектування вирішувалась в два кроки: визначення області інтересу на зображенні, і подальша локалізація та класифікація об'єктів всередині визначеного регіону.

Однокроковий підхід, запропонований Дж. Редмоном, дозволив значно скоротити час навчання та обробки зображень нейронними мережами детектування об'єктів та відкрив шлях до детектування в режимі реального часу. Після випуску оновлення YOLOv3, автор покинув роботу над проектом, обґрунтувавши це етичними міркуваннями щодо потенційного подвійного призначення свого алгоритму. Подальшим розвитком YOLO паралельно займається одразу декілька незалежних команд по всьому світу, тому назви версій відображають лише хронологію їх виходу, а не перевагу однієї над іншою. Наприклад, YOLOv10 може забезпечувати швидшу збіжність за YOLOv8, проте демонструвати нижчу точність при роботі з об'єктами малих розмірів.

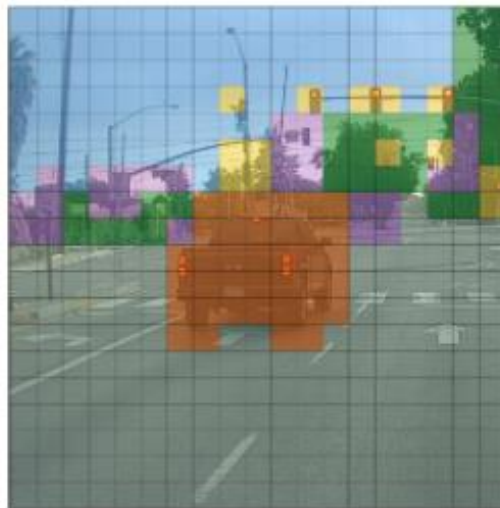


Рисунок 2.6 — Приклад обробки областей зображення алгоритмом YOLO

YOLO є однією з найбільш популярних моделей для детектування об'єктів, оскільки забезпечує оптимальний баланс між швидкістю обробки і точністю. Різні версії цієї моделі (YOLOv1 – YOLO11) внесли значні зміни та вдосконалення. Наприклад, YOLOv3 включає згорткову мережу Darknet-53 з проміжними з'єднаннями [23], а YOLOv4 [24] та YOLOv5 [25] оптимізували

⁶ Джерело зображення — <https://datascience.stackexchange.com/questions/26403/how-does-yolo-algorithm-detect-objects-if-the-grid-size-is-way-smaller-than-the>

роботу з ознаками на різних масштабах та автоматизували пошук оптимальних гіперпараметрів, що покращило точність і швидкість роботи. Версія YOLOv8 використовує повністю безякорне детектування та автоматизовані методи аугментації [26], що покращує ефективність обробки ознак. YOLOv9 [27] поєднує нові методи вдосконалення поширення інформації градієнта, а YOLOv10 [28] внесла покращення для зменшення затримки при детектуванні об'єктів та запропонувала альтернативний метод відсіювання рамок, що повторюються.

2.2.2 Алгоритмічні основи YOLO

YOLO працює за принципом розбиття зображення на сітку і одночасного прогнозування обмежувальних рамок об'єктів (англ. — bounding box) та їхніх класів для кожної клітинки [11].

Алгоритм починає роботу з розбиття вхідного зображення розміром $W \times H$ на сітку розміром $S \times S$, формуючи комірки (англ. cells) розміром $\frac{W}{S} \times \frac{H}{S}$. Кожна з комірок відповідає за визначення об'єктів, центри яких знаходяться всередині даної комірки, і прогнозує координати обмежувальної рамки — прямокутника, що описує місцезнаходження об'єкту на зображенні (задача регресії), та ймовірність його належності до одного з класів навчальної вибірки (задача класифікації). Вихід такої нейронної мережі можна позначити як вектор формату

$$y = [P_c, B_x, B_y, B_h, B_w, C_1, C_2, \dots, C_n],$$

де P_c позначає ймовірність знаходження об'єкта в комірці, B_x, B_y описують координати центру обмежувальної рамки відносно комірки, B_h, B_w

позначають висоту та ширину обмежувальної рамки відносно комірки, а $C_1, C_2, \dots, C_n$ це мітка класу серед n класів з навчальної вибірки (рис.).

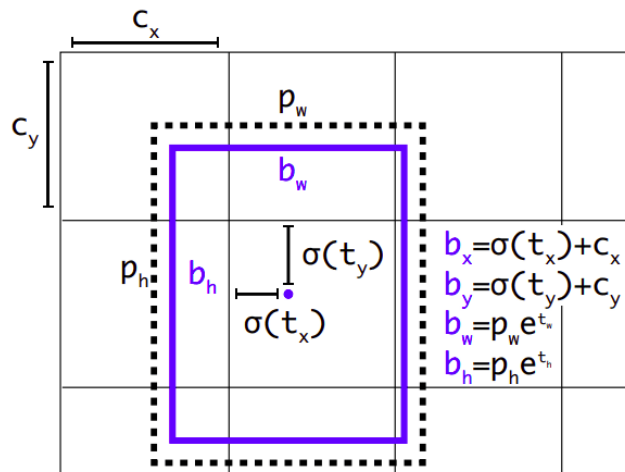


Рисунок 2.7 — Параметри обмежувальної рамки об'єкта, визначеної алгоритмом YOLO

Для оцінки відповідності прогнозованих обмежувальних рамок об'єкта істинним значенням з навчальної вибірки, в сімействі алгоритмів YOLO використовується метрика Intersection over Union (скор. IoU). Значення IoU обчислюється як відношення площі перетину двох рамок до суми їх площі:

$$IoU = \frac{Area_{intersection}}{Area_{union}},$$

де $Area_{intersection}$ — площа перетину, $Area_{union}$ — загальна площа обох рамок. При цьому, $IoU \in [0,1]$, де 1 позначатиме ідеальний збіг двох рамок.

Однак при прогнозуванні рамок об'єктів алгоритм може визначати одразу декілька різних рамок, що належатимуть одному і тому самому об'єкту. Для усунення надлишкових рамок в YOLO використовується алгоритм Non-Maximum Suppression (скор. — NMS), який гарантує, що для кожного з

⁷ Джерело зображення — https://www.researchgate.net/figure/YOLO-Bounding-Box-with-Location-Prediction-21_fig2_352086797

визначених об'єктів у фінальному прогнозі лишиться лише одна рамка, якій була призначена максимальна ймовірність. Алгоритм складається з декількох кроків: сортування прогнозованих рамок за зменшенням ймовірності їх належності об'єкту в порядку збування, видалення надлишкових рамок, для яких оцінена ймовірність не перевищує значення заданого користувачем мінімального порогу, та повторення цих кроків для решти прогнозованих рамок (рис. 2.8).

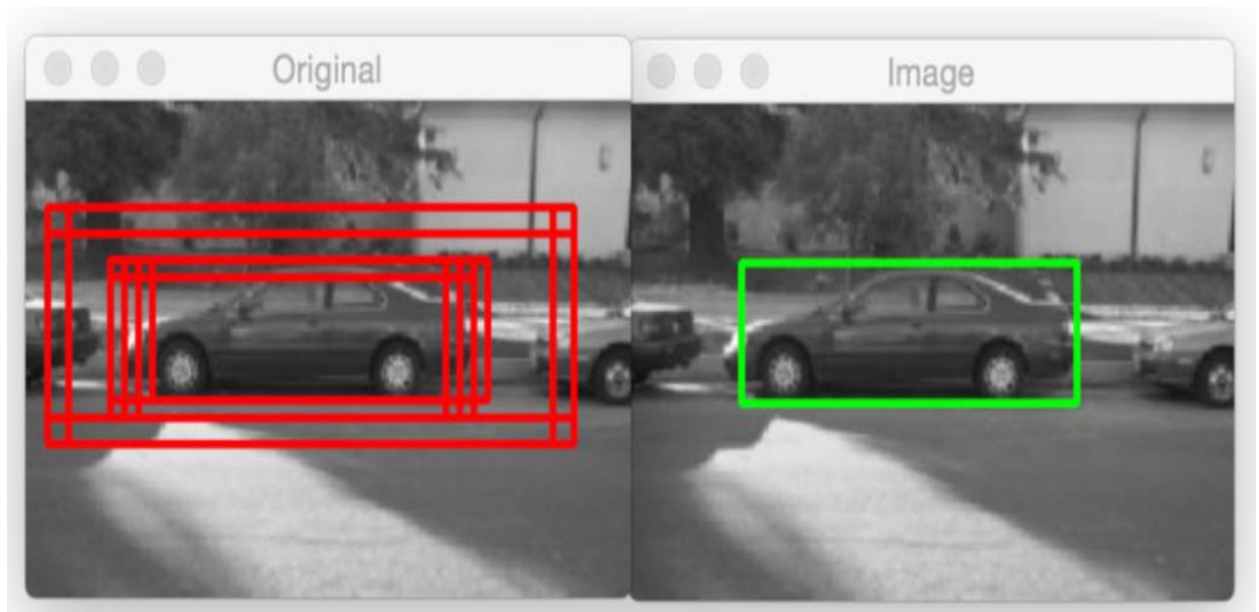


Рисунок 2.8 — Приклад роботи алгоритму Non-Maximum Suppression

Такий підхід довів свою ефективність, однак відома проблема хибно позитивних спрацювань алгоритму в сценаріях, коли цільові об'єкти є дрібними, щільно розташованими, або перекриваються один одним. Існують також модифікації даного методу, які замість жорсткого порогу відсіювання використовують ймовірнісну функцію (Soft NMS) [29], що дозволяє підвищити ефективність детектування щільно розташованих об'єктів, та модифікації, які враховують відстань від центру прогнозованої рамки до центру об'єкта (Distance-IoU NMS, скор. — DIoU-NMS) [30].

⁸ Джерело зображення — <https://cvexplained.wordpress.com/2020/07/15/2-8-non-maxima-suppression/>

Окрім алгоритму NMS, в деяких методологіях YOLO використовуються альтернативні підходи відсіювання надлишкових рамок, наприклад, алгоритм Dual Label Assignment, в якому для навчання моделі одночасно використовуються набір з декількох найкращих прогнозів для покращення точності при навчанні моделі і один конкретний прогноз в якості найкращого кандидата.

2.2.3 Топологія нейронних мереж на основі YOLO

Нейронні мережі на основі YOLO мають модульну архітектуру і складаються з трьох компонентів: згорткової нейронної мережі для вилучення ознак (англ. — backbone), модуля агрегації карт ознак (англ. — neck) та детектуючої голови (англ. — head). Поділ на частини є умовним, і може змінюватись залежно від версії методології, цільового призначення нейронної мережі та авторських модифікацій.

Backbone — це базова складова нейронних мереж на базі YOLO, що відповідає за вилучення ознак (англ. — features) зі вхідного зображення. В основі даного модуля лежить згорткова нейронна мережа, конкретна реалізація котрої залежить від версії YOLO. В сучасних модифікаціях YOLO, починаючи з YOLOv5, основними блоками backbone є згорткові блоки, блоки bottleneck та блоки агрегації контексту.

Згорткові блоки, що як Conv (рис. 2.9), вилучають локальні ознаки вхідного зображення та складаються з шарів згорки, нормалізації та активації. В якості нормалізації зазвичай застосовується операція батчевої нормалізації (англ. — batch normalization, BN), а в якості функції активації зазвичай застосовуються LeakyReLU або SiLU.

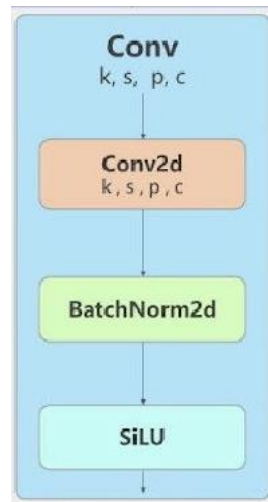


Рисунок 2.9 — Структура блоків Conv на прикладі мережі YOLOv8

Блоки bottleneck (рис. 2.10) дозволяють ефективно передавати інформацію про градієнт через глибоку нейронну мережу, зменшуючи обчислювальні витрати та запобігаючи проблемі затухання градієнта. Ідейним натхненням даних блоків є мережа ResNet, в якій вперше була втілена концепція залишкових з'єднань.

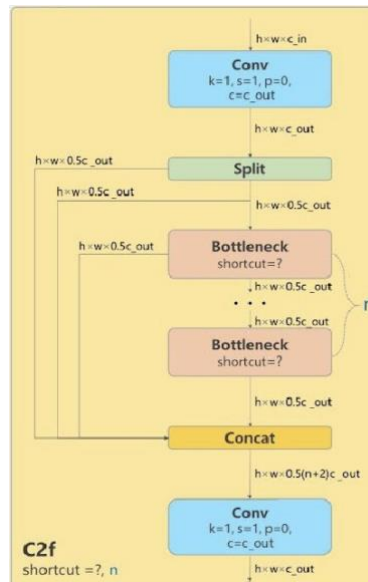


Рисунок 2.10 — Структура bottleneck-блоків на прикладі C2f

⁹ Джерело зображень — <https://viso.ai/deep-learning/yolov8-guide/>

Структура backbone-мережі містить також блок агрегації глобального контексту використовуються для поєднання інформації з різних масштабів карт ознак. Прикладом є Spatial Pyramid Pooling Fast (скор. — SPFF). Даний блок використовує серію згорток і операцій пулінга різних розмірів.

Neck — наступна складова детекторів об'єктів на основі YOLO, яка відповідає за агрегацію ознак різних рівнів (рис. 2.11). Ідея полягає в поєднанні різних масштабів карт ознак для отримання більш ефективної репрезентації об'єктів різних розмірів. Основними блоками neck-частини є оператори апсемплінгу (англ. — upsample), що використовуються для збільшення розміру карт ознак перед подальшим об'єднанням з картами ознак інших рівнів, та оператор конкатенації (англ. — concatenation) для власне реалізації об'єднання по каналах. За замовчуванням, апсемплінг зазвичай виконується шляхом інтерполяції, оскільки така операція не тягне за собою додаткових параметрів, однак може бути замінений більш ефективними операторами, наприклад, з модулями уваги або модулями врахування контексту. Також, у neck-частині мережі можуть застосовуватись згадані вище модулі bottleneck, для подальшої обробки об'єднаних ознак.

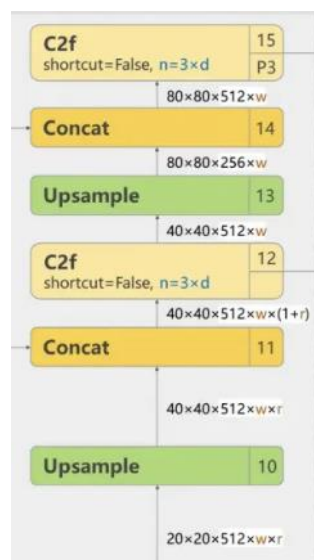


Рисунок 2.11 — Будова neck в YOLOv8

¹¹ Джерело зображення — <https://blog.roboflow.com/what-is-yolov8/>

Head — заключна частина мереж на основі методології детектування об'єктів YOLO (рис. 2.12), яка відповідає за прогнозування обмежувальних рамок об'єктів та ймовірності їх приналежності до класів з навчальної вибірки. В сучасних топологіях YOLO застосовується так звана роздвоєна голова (англ. — decoupled head), яка складається з модулю регресії (задача визначення рамок) та класифікації (оцінка ймовірності того, що об'єкт належить до певного класу).

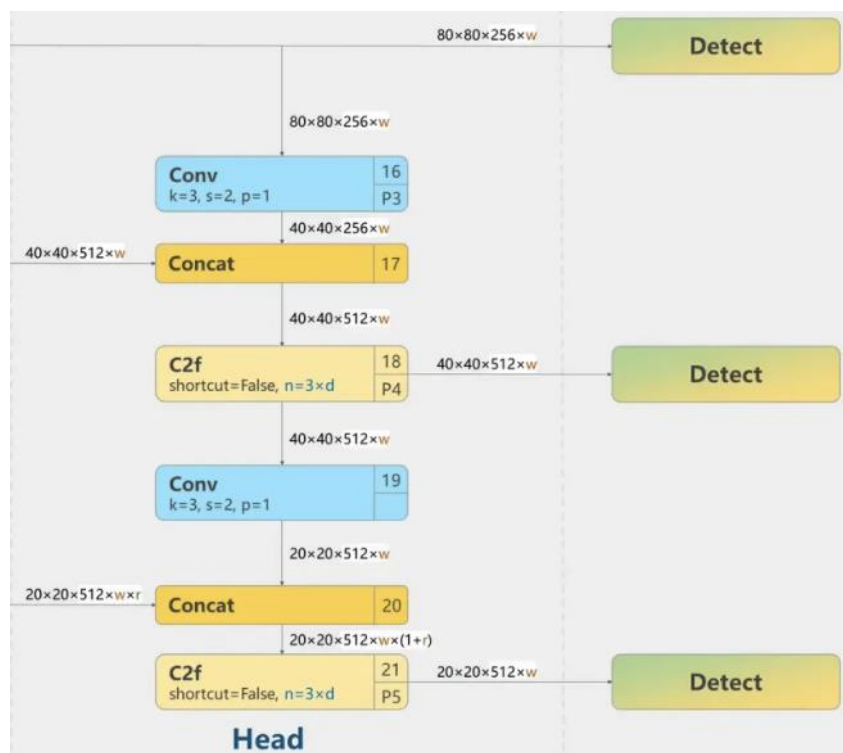


Рисунок 2.12 — Будова head в YOLOv8

Перевагами такої будови нейронної мережі є її модульність — детектори на базі YOLO можна адаптувати під конкретні задачі шляхом заміни окремих блоків мережі, та гнучкість — завдяки вилученню інформації про об'єкти на різних масштабах, мережу можна легко адаптувати під різні розміри цільових об'єктів.

¹² Джерело зображення — <https://blog.roboflow.com/what-is-yolov8/>

2.2.4 Теоретичні аспекти навчання мережі YOLO

Функція втрат нейронних мереж на базі методології YOLO, позначена $\mathcal{L}_{\text{YOLO}}$, складається з втрат регресії і втрат класифікації. В нових версіях YOLO, CIOU [31] використовується як функція втрат регресії рамок об'єктів. Дана функція визначається як:

$$\mathcal{L}_{\text{CIOU}}(b, \hat{b}) = 1 - \text{IoU}(b, \hat{b}) + \frac{\rho^2(b_c, \hat{b}_c)}{c^2} + \alpha \cdot v,$$

де $b, \hat{b}$ це істинні і передбачені координати обмежувальних рамок об'єктів, відповідно, $\text{IoU}(b, \hat{b})$ це коефіцієнт перекриття (англ. — Intersection over Union), який відображає площу перекриття між передбаченими та істинними рамками:

$$\text{IoU}(b, \hat{b}) = \frac{|b \cap \hat{b}|}{|b \cup \hat{b}|},$$

де $|b \cap \hat{b}|$ це площа перекриття, $|b \cup \hat{b}|$ це загальна площа, яку перекривають обидві рамки, $\rho(b_c, \hat{b}_c)$ це Евклідова відстань між центроїдами $b_c, \hat{b}_c$, α це коефіцієнт ваги, що визначає вплив схожості між співвідношенням сторін передбачених та істинних рамок:

$$\alpha = \frac{v}{(1 - \text{IoU}(b, \hat{b})) + v},$$

v позначає дану схожість між співвідношенням сторін передбачених та істинних рамок:

$$v = \frac{4}{\pi^2} \left(\arctan \left(\frac{w}{h} \right) - \arctan \left(\frac{\hat{w}}{\hat{h}} \right) \right)^2,$$

де w та h це ширина та висота істинної рамки об'єкта, а $\hat{w}, \hat{h}$ — ширина й висота передбаченої рамки для нього.

За задачу класифікації відповідає варіфокальна функція витрат (англ. — varifocal loss function, VFL) [32], що визначається як:

$$\mathcal{L}_{\text{VFL}} = - \sum_{i=1}^N (\alpha y_i (1 - \hat{p}_i)^\gamma \log(\hat{p}_i) + (1 - y_i) \hat{p}_i^\gamma \log(1 - \hat{p}_i)),$$

де y_i це істинна мітка класу, $\hat{p}_i$ вектор ймовірності передбачених класів, α коефіцієнт ваги, а γ коефіцієнт регуляризації.

Загальна функція витрат нейронних мереж визначається як сума двох зазначених вище функцій витрат і фокальних втрат, помножених на відповідні вагові коефіцієнти:

$$\mathcal{L}_{\text{YOLO}} = \lambda_{\text{CIoU}} \mathcal{L}_{\text{CIoU}} + \lambda_{\text{VFL}} \mathcal{L}_{\text{VFL}} + \lambda_{\text{DFL}} \mathcal{L}_{\text{DFL}},$$

де $\lambda_{\text{CIoU}}, \lambda_{\text{VFL}}, \lambda_{\text{DFL}}$ це гіперпараметри, що зважують важливість кожної з функцій у рамках загальної функції витрат.

Функції витрат нейронної мережі оптимізується алгоритмом стохастичного градієнтного спуску (англ. — Stochastic Gradient Descent, SGD) [33]. Правило оновлення вагів на кожній ітерації навчання можна сформулювати як:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \mathcal{L},$$

де η це коефіцієнт швидкості навчання, що керує розміром кроку в протилежну від градієнта сторону, а $\nabla_{\theta} \mathcal{L}$ це градієнт функції витрат відповідно до параметрів моделі.

2.3 Методи компресії нейронних мереж на базі YOLO

Для розміщення мережі YOLO на пристроях з обмеженими обчислювальними ресурсами та автономністю була розроблена низка методів, що сприяють зниженню розміру та обчислювальної складності мережі. В контексті застосування даної топології до задачі детектування підводних об'єктів, можна виділити наступні методи компресії.

Урізання (англ. — pruning) — передбачає видалення з мережі елементів, які несуть мінімальний вплив на вихідний результат, а отже можуть бути усунені без значного погіршення продуктивності.

Вперше техніка pruning була описана в статтях до методів Optimal Brain Damage та Optimal Brain Surgeon [34]. В ранніх версіях алгоритмів, для оцінки важливості параметрів використовувалось розкладання в ряд Тейлора, оскільки оцінка кожного параметру окремо займає довгу кількість часу. Пізніше, для оцінки важливості параметрів стали застосовуватись інші методи, такі як ℓ_n -норма, оцінка активації карт ознак, фактор масштабування батчевої нормалізації, тощо. Існують також методи структурованого урізання, коли оцінюється вплив цілої структури в складі нейронної мережі, наприклад, шару або блоку

Квантизація (англ. — quantization) — полягає у репрезентації ваг нейронної мережі з меншою чисельною точністю, аніж оригінальні значення. Це дозволяє використовувати для обчислень окремі блоки в складі апаратного забезпечення, які оптимізовані для роботи з певним форматом чисел (наприклад, int8).

При виборі методів квантизації мережі враховується вплив зниження чисельної точності на фінальну точність нейронної мережі. Квантизація може відбуватись як в процесі навчання (Quantization Aware Training, скор. — QAT), коли чисельна точність ваг зменшується після кожного оновлення градієнтів, так і після навчання (Post Training Quantization, скор. — PQT). Останній метод

є менш витратним обчислювально, однак результуюча точність нейронної мережі значно знижується порівняно з квантизацією в процесі навчання, тому метод використовується рідше.

2.4 Гібридні моделі детектування підводних об'єктів

2.4.1 Загальна характеристика

Для подолання обмежень, викликаних низькою якістю підводних зображень, особливостями підводних об'єктів таких як малий розмір і щільне розташування, а також обчислювальними обмеженнями, розробляють спеціалізовані підходи до детектування підводних об'єктів. Часто дані підходи ґрунтуються на використанні гібридних нейронних мереж, коли до відомої топології, наприклад, YOLO, додаються окремі модулі або блоки, спрямовані на підвищення ефективності роботи у конкретних сценаріях. Такі гібридні нейронні мережі можна розділити на три основні категорії за способом, яким досягається підвищення ефективності.

Перша категорія включає детектори з покращеною репрезентацією ознак, що досягається шляхом збільшення глибини мережі або додавання додаткових блоків. Наприклад, механізми уваги часто використовуються для посилення здатності моделі виділяти важливі ознаки, на які слід зосередитися в процесі обробки. Також деякі гібридні мережі використовують блоки Transformer в складі backbone або neck частин детектора, припускаючи, що гетерогенні архітектури позитивно впливають на варіативність вилучених ознак. Інші методології зосереджуються на збільшенні якості та кількості даних шляхом їх аугментації та попередньої обробки для досягнення вищої точності детектування. Проблемою таких підходів є підвищена обчислювальна складність мережі, оскільки збільшена глибина мережі та наявність додаткових блоків можуть покращувати точність, але ускладнюють

застосування в режимі реального часу та на обмежених за потужністю пристроях через високі затримки та велику кількість параметрів. Отже, такі моделі можуть застосовуватись лише на стаціонарному обладнанні на попередньо зібраних даних.

Друга категорія фокусується на підвищенні точності через зменшення впливу конкретних проблем підводних зображень, таких як малий розмір об'єктів, їхнє перекриття та розмитість. Зазвичай застосовуються методи вдосконалення апсемплінгу ознак, додаткові модулі для виявлення дрібних об'єктів або такі блоки, як модулі уваги (англ. — attention). Основна проблема цих методів полягає в їхній обмеженій узагальнюваності, адже покращення, досягнуті для певного набору даних, можуть не застосовуватися до інших наборів, обмежуючи їхню практичну цінність.

Третя категорія зосереджується на зниженні кількості параметрів, розміру і затримки при обробці зображень. Стиснення зазвичай здійснюється через модифікацію backbone-мережі, відповідальної за вилучення ознак. Наприклад, в детекторах на основі YOLO ця частина відповідає за понад 50% загальної обчислювальної складності. Backbone частина детектора часто замінюється на мобільні або легкі нейронні мережі, такі як FasterNet [35], GhostNet [36] або інших, що суттєво зменшує розмір моделі та час збіжності, роблячи її придатною для використання на пристроях з обмеженими ресурсами, до яких відноситься інтегроване обладнання автономних підводних апаратах. Однак у цьому випадку досягнуті покращення у швидкості та розмірі тягнуть за собою зменшення потужності мережі, що знижує точність у порівнянні з загальновідомими алгоритмами. Для усунення цього ефекту часто додають додаткові шари або обчислювальні блоки для досягнення компромісу між обчислювальною складністю та точністю нейронної мережі.

деформовані згортки (англ. – deformable convolutions), що дозволило зменшити обчислювальну складність запропонованої мережі. Однак для того щоб компенсувати негативний ефект на точність, спричинений використанням мобільної архітектури згорткової мережі, автори додали модулі уваги для покращеної репрезентації ознак на етапі їх агрегації та окрему детектуючу голову для малих об'єктів (рис 2.14).

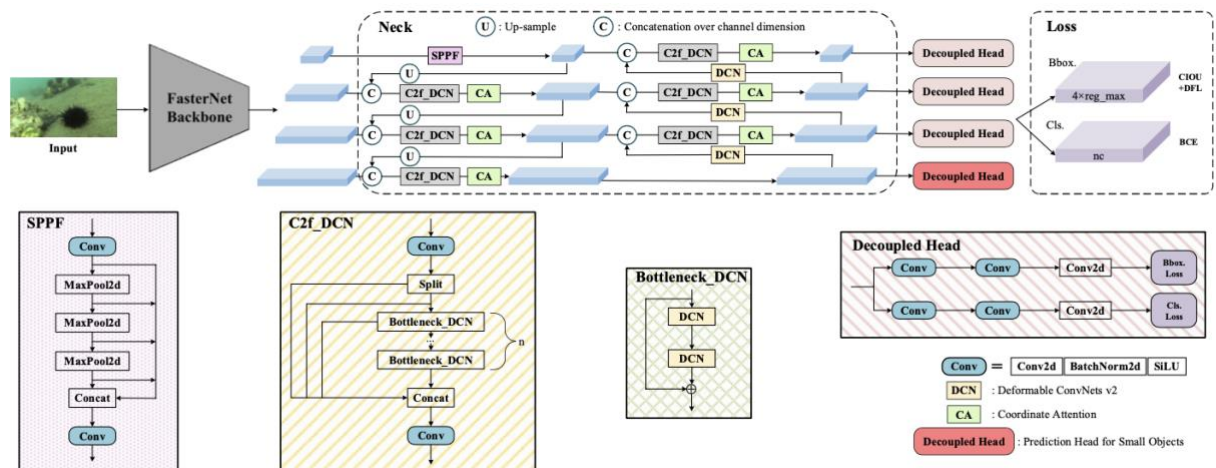


Рисунок 2.14 — Топологія гібридної нейронної мережі на базі FasterNet, призначеної для детектування підводних об'єктів

Розробці підходів до створення гібридних нейронних мереж для детектування підводних об'єктів присвячена ще низка робіт. Зокрема, у роботі [39] розглянуто гібридну мережу з архітектурою backbone-мережі від MobileNetV2 і ядрами згортки depthwise separable convolution для зменшення обчислювальної складності. При цьому, втрату точності, спричинену компресією мережі, було компенсовано введенням авторського Attentional Feature Fusion, що додає операцію уваги у процес агрегації карт ознак різного масштабу.

У статті [40] автори роботи застосували ряд оптимізацій для розміщення нейронної мережі на малопотужному одноплатному комп'ютері NVIDIA

Jetson Nano, застосувавши ядра згортки розміром 1×1 в backbone-мережі та модулі уваги FPN-Attention для агрегатора ознак, що дозволило проводити обробку зображень з частотою 9 кадрів на секунду.

В дослідженні [41] було застосовано модифіковану топологію мережі YOLOv7, оптимізовану для виявлення малих за розміром об'єктів шляхом додавання модуля уваги Convolutional Bottleneck Attention Module (скор. — CBAM) для збагачення карт ознак додатковою семантичною інформацією.

Механізми уваги та методи їх інтеграції в процес детектування на базі карт ознак різних масштабів були розглянуті також у статті [42]. В якості основи топології була обрана мережа YOLOv5, а цільовою задачею була обрана оцінка біорізноманіття підводного середовища на базі вибірки, що складалась з чотирьох класів морських організмів.

В дослідженні [43] для покращення ефективності детектування у підводному середовищі були застосовані методи з ретинекс-теорії. Зображення покращеної якості з усунутими дефектами освітлення подавались на вхід нейронної мережі. Результати експериментів довели, що такий підхід забезпечує вищу точність, порівняно з оригінальною топологією YOLOv4.

Висновки до розділу 2

Розділ присвячено огляду методів попередньої обробки зображень та детальному розгляді алгоритмічної основи методів You Only Look Once (скор. — YOLO) для задач детектування об'єктів, історію їхнього розвитку, методи компресії нейронних мереж даної архітектури. Детально досліджено топологію мереж на базі YOLO та наведена низка потенційних можливостей до вдосконалення існуючих методів для оптимальної ефективності в контексті поставленої задачі.

Розглянуті гібридні мережі детектування підводних об'єктів, що оптимізовані для виконання задач виявлення цілей та класифікації в умовах поганої видимості або обмежених обчислювальних потужностей. В результаті встановлено, що можна виділити три види таких гібридних нейронних мереж: детектори з покращеною репрезентацією ознак, що досягається шляхом збільшення глибини мережі або додавання додаткових блоків; детектори, націлені на підвищенні точності через зменшення впливу конкретних проблем підводних зображень, таких як малий розмір об'єктів, їхнє перекриття та розмитість; і такі, що зосереджується на зниженні кількості параметрів, розміру і затримки при обробці зображень. Оглянуто конкретні приклади таких гібридних моделей, виділено їх сильні та слабкі сторони.

РОЗДІЛ 3 ІНТЕЛЕКТУАЛЬНА СИСТЕМА ОБРОБКИ ЗОБРАЖЕНЬ НА АВТОНОМНИХ ПІДВОДНИХ АПАРАТАХ

Для розв’язання задачі детектування об’єктів в підводному середовищі передбачається розробка програмного забезпечення, яке має відповідати наступним критеріям:

- можливість ефективного виявлення і класифікації об’єктів інтересу в підводному середовищі з мінімальною кількістю хибно-негативних і хибно-позитивних спрацювань;
- здатність до роботи в умовах низької якості зображення та при наявності оптичних перешкод, таких як недостатня або надмірна освітленість знімка або його частин, низка локальна або глобальна контрастність знімка, наявність візуальних артефактів;
- висока точність при детектуванні об’єктів, що частково приховані за іншими об’єктами або щільно розташовані відносно один одного, а також можливість визначення малих за розміром об’єктів;
- висока обчислювальна ефективність та можливість розміщення програмного забезпечення на окремі модулі, такі як одноплатні комп’ютери серії NVIDIA Jetson та Raspberry Pi.

Для реалізації програмного забезпечення передбачено побудову наступних програмних модулів:

- модуль попередньої обробки зображень для усунення оптичних перешкод, нормалізації контрастності знімку та приведення зображення до розміру й характеристик для подання на вхід нейронної мережі;
- нейронна мережа детектування об’єктів на базі модифікованої архітектури YOLOv8 з використанням власних методів компресії згорткової (backbone) мережі та покращеннями для роботи з об’єктами малого розміру та об’єктами, що перекриваються;

- клієнтська частина програмного забезпечення, яка дозволяє імпортувати модель YOLO у форматі .onnx або .pt та візуалізувати результати детектування об'єктів.

3.1 Методологія детектування підводних об'єктів

3.1.1 Вибір методів попередньої обробки

Попередня обробка починається з приведення розміру зображення до такого, який можна подати на вхід нейронної мережі. Оскільки використовуються моделі YOLO, розмірність зображення має становити 640 x 640 x 3 пікселі. Попередній аналіз навчальних вибірок показав, що в них немає зображень меншого розміру, тому не звертаємось до технік збільшення роздільної здатності.

Виконується аугментація даних для більш стабільного процесу навчання та розширення навчальної вибірки зразками об'єктів під різними кутами та нахилами, а також накладанням деформацій, які імітуватимуть реальне середовище.

Бібліотека Ultralytics містить реалізації ряду аугментацій, таких як випадкові повороти; випадкове кадрування; деформації перспективи зображення; випадкове відзеркалення; мозаїчна аугментація; випадкове накладання об'єктів з інших зображень вибірки, їх повороти та відзеркалення; аугментації, що стосуються випадкових змін яскравості, насиченості та кольорового балансу зображення; методи аугментації з методології RandAugment.

Для подолання проблем низької контрастності підводних зображень та нерівномірного освітлення, було розроблено модуль корекції характеристик зображення на базі методу контраст-обмеженої гістограмної еквалізації. Вибір алгоритму ґрунтувався на проведених експериментах щодо часу обробки

одиночного зображення та впливу такої обробки на показники точності нейронної мережі. Заміри метрик проводились на топології мережі YOLOv8s без авторських модифікацій. Результати наведено в таблиці 3.1.

Таблиця 3.1 — Дослідження впливу алгоритмів попередньої обробки на швидкість та точність мережі

	<i>Алгоритм</i>	<i>Значення precision</i>	<i>Значення recall</i>	<i>Час обробки зображення (мс)</i>
1	-	0.8273	0.7896	-
2	Гістограмна еквалізація (HE)	0.808	0.7667	2.5
3	Адаптивна гістограмна еквалізація (AHE)	0.7867	0.7522	2.7
4	Контраст-обмежена адаптивна гістограмна еквалізація (CLAHE)	0.8267	0.8102	2.5

Виявлено, що контраст-обмежена адаптивна гістограмна еквалізація надає найкращий ефект щодо якості відновлення зображення (рис. 3.1), при цьому додаючи до 3 мілісекунд на обробку кожного зображення.



Рисунок 3.1 — Алгоритм CLAHE, застосований до зображення з вибірки

Результати показують, що застосування даного алгоритму для попередньої обробки дозволяє скоротити кількість пропущених об’єктів,

ймовірно, за рахунок покращеної видимості деталей на затемнених ділянках зображення.

Алгоритм багатошкальної ретинекс-корекції з відновленням кольору, що розглядався в теоретичній частині роботи, не було включено до порівняння через значний час обробки кожного зображення, що перевищував 1 секунду.

3.1.2 Визначення топології нейронної мережі

Для досягнення оптимального балансу між обчислювальною ефективністю і точністю в задачі детектування об'єктів у підводному середовищі було розроблено топологію нейронної мережі на базі YOLOv8. Для досягнення поставлених задач було запропоновано декілька модифікацій оригінальної архітектури:

- застосувати методи компресії нейронної мережі детектування об'єктів для зменшення обчислювальних витрат;
- ввести зміни у процес агрегації ознак шляхом додавання додаткових модулів або модифікувати процес апсемплінгу карт ознак в писк-частині детектора для збереження семантичної інформації, що покращить ефективність мережі при визначенні малих цілей та об'єктів неправильної форми;
- змінити алгоритм відсіювання прогнозованих рамок з жорстким порогом відсіювання (Distance-IoU NMS) на алгоритм Soft NMS, який використовує ймовірнісну функцію оцінювання замість жорсткого порогу і зменшить кількість хибно-негативних спрацювань у сценаріях з щільним розташуванням об'єктів та випадками часткового їх перекриття.

Оскільки операції згортки відповідають за більше ніж половину загальної обчислювальної складності мережі YOLOv8, в якості методу

зменшення розміру нейронної мережі запропоновано модифікувати згорткові та bottleneck-блоки мережі використовуючи структури, що застосовуються в мобільних нейронних мережах.

За полегшену реалізацію згорткової операції було обрано мережу GhostNet. Принцип її роботи полягає в генерації карт ознак за допомогою додаткового згорткового ядра меншого розміру і подальшої генерації на їх основі додаткових карт ознак за допомогою обчислювально недорогих операцій, наприклад, згортки з фільтром 1×1 (рис. 3.2).

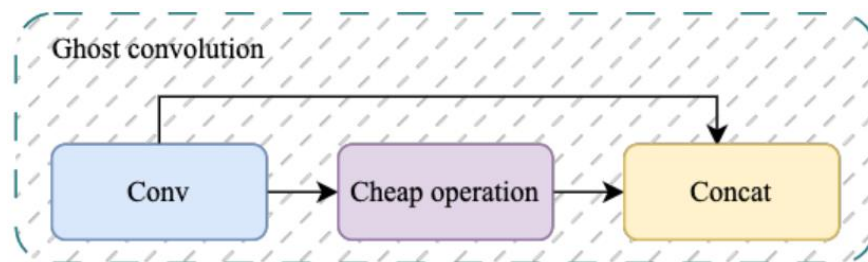


Рисунок 3.2 — Схема згорткової операції з мережі GhostNet

На основі даної операції можна вивести bottleneck-блок з меншою кількістю параметрів (рис. 3.3).

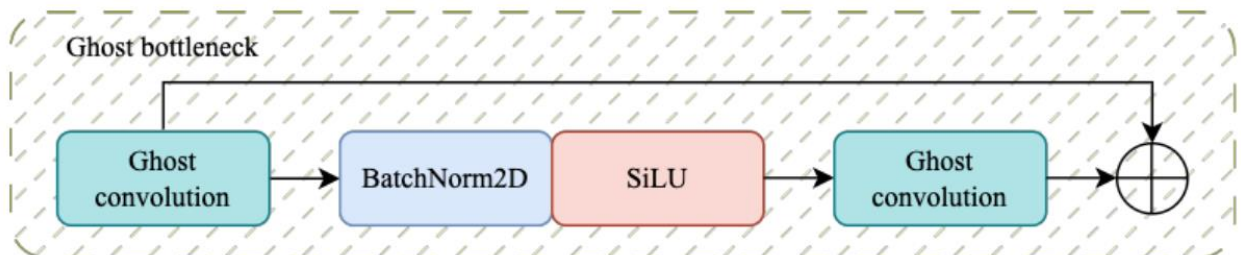


Рисунок 3.3 — Bottleneck-блок на базі операції згорки з GhostNet

Отриманий блок можна використовувати у складі модулів C3 та C2f з оригінальної архітектури YOLOv8 для скорочення кількості обчислень за

рахунок використання менших ядер згортки при збереженні схожого рівня семантичної інформації (рис. 3.4).

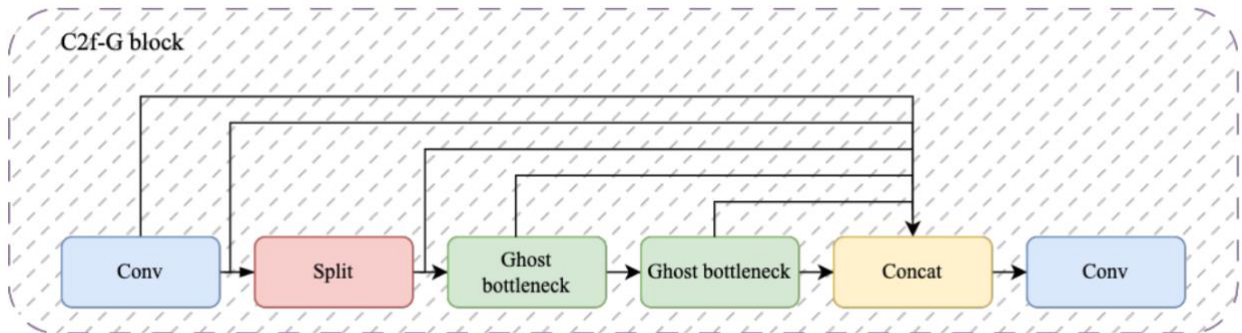


Рисунок 3.4 — Блок C2f-G з використанням Ghost bottleneck

Виведений блок може слугувати заміною для модулів C2f з оригінальної архітектури YOLOv8. Через їх положення в загальній архітектурі мережі можна регулювати обчислювальну складність мережі — розміщення модулю C2f-G в backbone буде мати найбільший ефект на кількість параметрів, однак якість отриманих карт ознак буде меншою за C2f. Натомість, розмістивши блок C2f-G в neck-частині детектора, позитивний ефект від зменшення обчислювальної складності все ще зберігається, але якість карт ознак практично не зазнає змін [44].

В агрегації ознак запропоновано змінити операцію інтерполяції на модуль Content-Aware ReAssembly of FEatures (скор. — CARAFE), який виконує операцію апсемплінгу з урахуванням вмісту. CARAFE працює в два основних етапи: генерація ядер перезбірки зображення з урахуванням контенту за допомогою структури зі згортками 1 x 1, енкодера для збереження семантичної інформації і операції нормалізації ядра для утримання суми значень ядер в межах одиниці. Згодом, модуль перезбірки виконує зворотню операцію і відтворює оригінальну карту ознак в збільшеному розмірі, враховуючи її вміст, і таким чином, зберігаючи і підсилюючи семантичну інформацію. Загальна структура мережі показана на рис. 3.5.

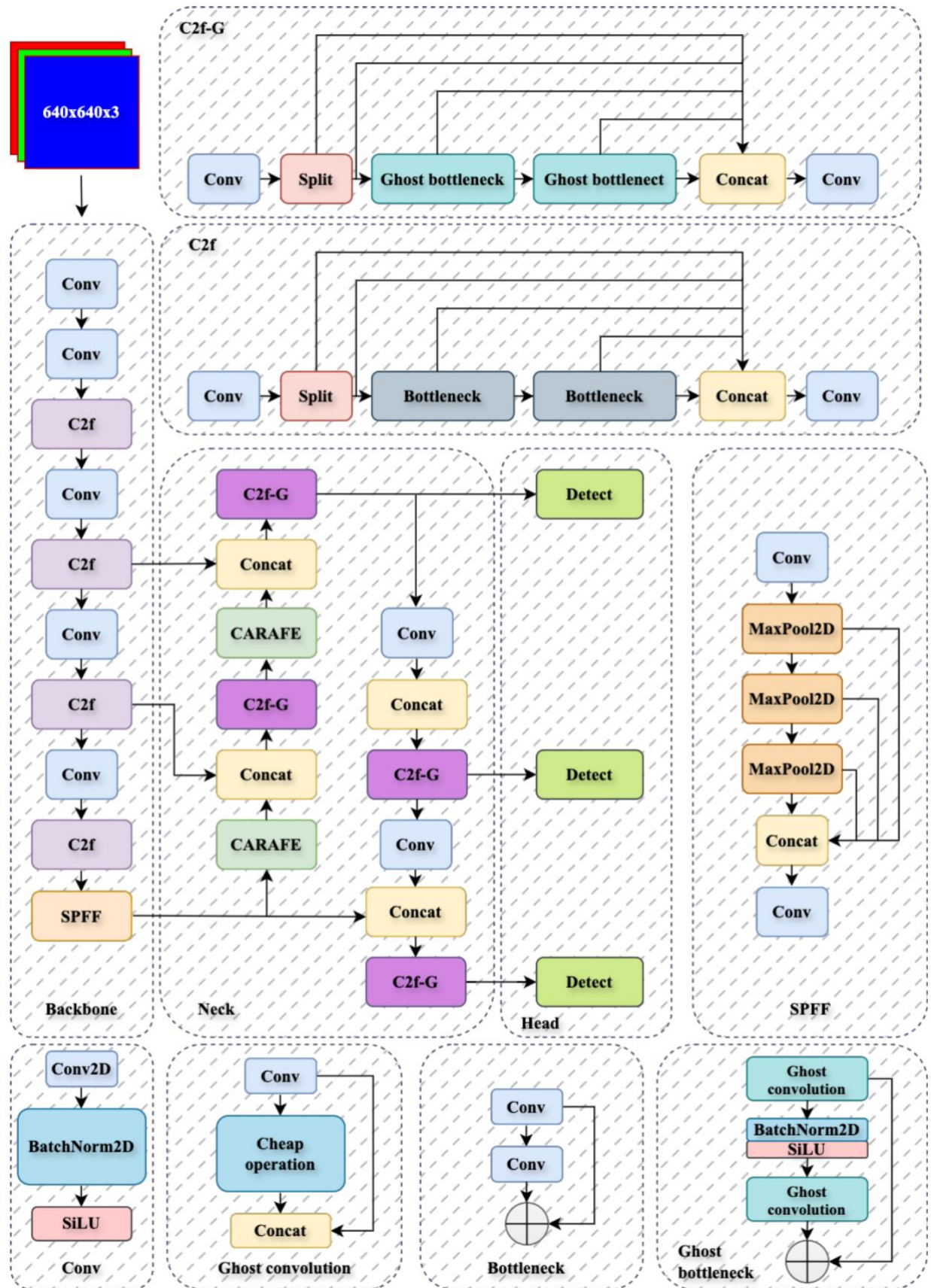


Рисунок 3.5 — Загальна структура запропонованої мережі

Окрім цього, пропонується використати алгоритм Soft Non-Maximum Suppression (скор. — Soft NMS) для відсіву прогнозованих рамок, які ймовірно належать одному й тому ж об'єкту. На відміну від традиційного алгоритму Non-Maximum Suppression, який використовує жорсткий поріг для відсіву дублюючих рамок або його модифікацій з урахуванням відстані між центрами рамок, «м'яка» версія алгоритму застосовує ймовірнісну функцію, що дозволяє оптимізувати мережу для визначення щільно розташованих об'єктів.

3.2 Вибір технологій і середовища розробки

Побудову нейронних мереж можна реалізувати з використанням різних мови програмування, кожна з яких має свої особливості, переваги та недоліки. Найбільш популярною мовою в сфері глибокого навчання є **Python**, більшість наявних фреймворків для навчання глибоких мереж написана саме цією мовою. Python забезпечує зручність і швидкість розробки завдяки великій кількості існуючих бібліотек, таких як TensorFlow, PyTorch і Keras, що надають готовий інструментарій для побудови, синтезу та тестування нейронних мереж. Головними перевагами мови Python є простота синтаксису, активна спільнота та велика кількість навчальних матеріалів. Недоліком є відносно низька швидкість виконання коду порівняно з мовами на основі компіляції. Це може стати критичним для задач, які мають певні обмеження для використання обчислювальних ресурсів.

C та C++ також широко використовуються у нейронних мережах, особливо для задач, що потребують високої обчислювальної ефективності, зокрема на вбудованих пристроях. Бібліотеки, такі як Caffe чи TensorRT, надають можливість ефективної реалізації моделей на нижчому рівні коду. Основні переваги C++ включають високу швидкість виконання та можливість тонкого налаштування алгоритмів. Однак дана мова має складніший

синтаксис і слабші можливості інтеграції зі сторонніми системами, ніж Python. Слід також зауважити про те, що основний код програмного забезпечення можна реалізувати на Python, а критичні частини коду, що вимагають швидшого виконання чи прямих звернень до апаратних пристроїв, написати мовою C++.

Мова **R** відома наявністю потужних інструментів для статистики та візуалізації даних, тому вона також широко використовується в машинному навчанні. Ця мова підходить для побудови математичних моделей та аналізу даних, проте її продуктивність поступаєтьс Python і C++, а менша кількість бібліотек глибокого навчання робить її гіршим вибором в контексті поставленої задачі.

Julia — це сучасна мова програмування, розроблена спеціально для високопродуктивних обчислень, що робить її перспективним вибором для машинного навчання та побудови нейронних мереж. Основною метою створення Julia було об'єднання швидкості низькорівневих мов програмування, таких як C і Fortran, із простотою та зручністю мов високого рівня, таких як Python або MATLAB. Важливою перевагою мови порівняно з Python є висока продуктивність завдяки компіляції коду JIT-компілятором, що дозволяє Julia працювати майже з такою ж швидкістю, як C++. Окрім цього, Julia підтримує багатопоточність і розподілені обчислення, що полегшує паралельну обробку великих обсягів даних. Для мови існують готові бібліотеки для побудови нейронних мереж, найбільш відомою є Flux.jl. Однак загалом, екосистема модулів і кількість ресурсів для підтримки та навчання в Julia значно поступаються Python.

Зважаючи на даний порівняльний аналіз, для розробки програмного забезпечення було обрано мову програмування Python. Розглянемо також бібліотеки глибокого навчання, доступні в екосистемі Python.

TensorFlow — одна з найпопулярніших бібліотек для глибокого навчання, розроблена компанією Google. Бібліотека підтримує побудову, тренування та розгортання нейронних мереж та має низку вбудованих методів

для оптимізації обчислень на графічних процесорах. Однією з ключових особливостей TensorFlow є робота з обчислювальними графами та розвинена екосистема, що включає утиліти для візуалізації метрик навчання і тестування мережі, додатки LiteRT та TensorFlow.js для розгортання моделей на мобільних пристроях та в браузері.

Keras — це високорівнева бібліотека для глибокого навчання, спрямована на спрощення процесу створення та навчання нейронних мереж завдяки простому синтаксису. Keras забезпечує модульність і гнучкість, що дозволяє легко комбінувати різні шари, функції активації, оптимізатори та інші компоненти мережі. Окрім цього, Keras дозволяє створювати функціональні моделі зі складними графами нейронних мереж.

PyTorch — це бібліотека для глибокого навчання, розроблена Facebook, що використовує динамічні обчислювальні графи. Також бібліотека містить інструменти для автоматичного диференціювання (Autograd), що спрощує реалізацію складних архітектур. PyTorch також відомий розвиненими інструментами для обчислень на графічних процесорах, що є особливо актуальним для моделей комп'ютерного зору.

Іншим важливим аспектом є вибір середовища для навчання нейронної мережі. Оскільки в роботі вирішується задача детектування об'єктів, мова йде про великі вибірки даних, які вимірюються тисячами зображень. Такі обсяги обчислень можуть ефективно проводитись лише на графічних процесорах. Існує декілька варіантів середовищ: локальне, на відкритих платформах або на орендованих потужностях графічних процесорів. Розглянемо кожен з варіантів.

Локальне середовище передбачає використання персонального апаратного забезпечення. Перевагою є повний контроль та конфіденційність, оскільки користувач в такому разі має повний доступ до налаштувань системи, версій бібліотек та інструментів, а дані не відправляються на сторонні сервери. Однак це вимагає значних фінансових витрат на апаратне забезпечення і його обслуговування. Також при такому навчанні слід забезпечити безперервне

живлення системи або можливість періодичного збереження ваг нейронної мережі для продовження навчання після збоїв.

Відкриті платформи, такі як Google Colab або Kaggle пропонують безкоштовний або умовно-безкоштовний доступ до обчислювальних ресурсів у хмарі. Ці ресурси є простими у використанні, оскільки бібліотеки та середовище розробки вже встановлені на хмарне апаратне забезпечення. Код зберігається на серверах компаній, доступ до нього можна отримати з будь-якого пристрою. Однак ресурси в таких сервісах обмежуються за часом та доступністю - зазвичай одна сесія роботи може тривати до 12 годин, дані після закінчення сесії або розриву з'єднання не зберігаються. Також такі сервіси не підходять для роботи з конфіденційною інформацією.

Орендовані потужності GPU, такі як надаються AWS, Azure, Google Cloud є альтернативним варіантом навчати нейронні мережі в хмарі. Вони дозволяють масштабувати ресурси і розподіляти обчислення між декількома апаратними пристроями, є гнучкими і забезпечують корпоративні стандарти захисту інформації та безперебійної роботи. Вартість таких сервісів є достатньо високою, а доступність може залежати від регіону, в якому перебуває користувач.

Підсумовуючи, для побудови та навчання нейронної мережі було обране локальне середовище, мову програмування Python та фреймворк глибокого навчання PyTorch.

3.3 Навчальні вибірки

Для навчання моделей та оцінки їх ефективності була застосована вибірка підводних зображень Underwater Target Detection Algorithm Competition 2020, яка складається зі знімків біологічних організмів у океанічному середовищі.

Навчальна вибірка складається з 5168 зображень в тренувальному наборі та 1293 зображення для валідації. Роздільна здатність знімків коливається від 586 x 480 до 3840 x 2160, медіанне значення — 1920 x 1080. Загальна кількість анотацій у вибірці — 46 680. На зображеннях представлено чотири класи:

- *echinus* — морський їжак, об’єкти даного класу характеризується невеликим розміром і складною формою контурів, часто є щільно розташованими відносно один одного і присутні у великій кількості на одному зображенні. Всього представлено 24 501 об’єктами у вибірці;
- *holothurian* — морський огірок, на зображеннях часто не виділяються на фоні піску та мулу, тому є складними для детектування. Є найменш репрезентованим класом у вибірці. Представлені 6 142 об’єктами;
- *scallop* — молюски, мають малий розмір на зображеннях, слабо виділяються на складному фоні. Незбалансований клас, представлений 7 250 об’єктами;
- *starfish* — морська зірка, виділяється за кольором і розміром, проте представлена 8 787 об’єктами у вибірці.

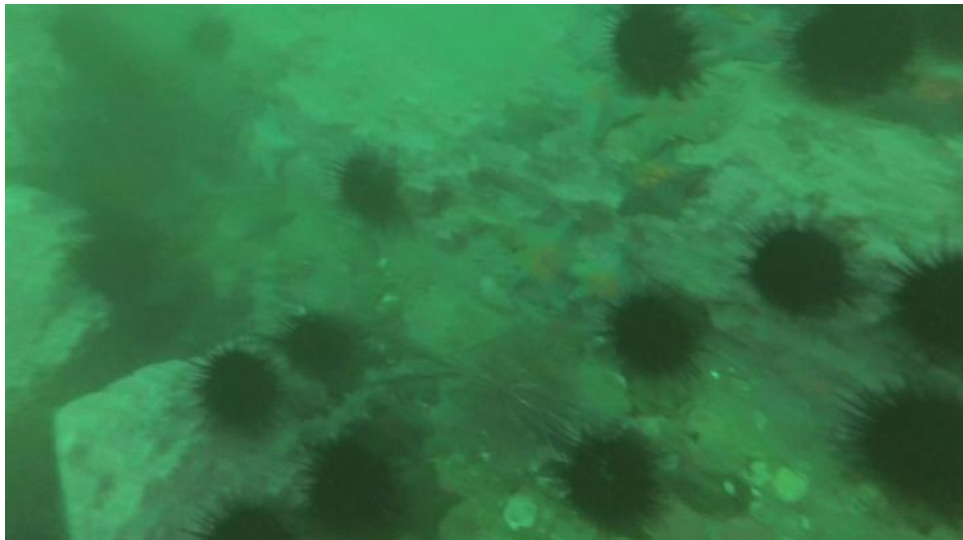


Рисунок 3.6 — Приклад зображення з валідаційної вибірки UTDAC 2020

Окрім цього, для валідації результатів в якості альтернативної вибірки було обрано додаткову вибірку Underwater Robot Picking Contest 2019 (скор. — URPC2019), яка включає ті самі класи і представлена 3765 навчальними та 942 тестовими зображеннями.

3.4 Експериментальні дослідження

3.4.1 Метрики для оцінки ефективності мережі

В експериментальних дослідженнях було використано наступний набір метрик:

- mAP_{50} — Mean Average Precision при значенні Intersection over Union 0.5, використовується для загальної оцінки точності мережі в локалізації й класифікації об’єктів, більше означає краще;
- mAP — Mean Average Precision при значенні IoU 0.50:0.05:0.95, оцінює точність в локалізації об’єктів, більше означає краще;
- параметри — кількість параметрів нейронної мережі, вказує на обчислювальну складність, менше означає краще;
- FLOPs — кількість операцій з рухомою комою на секунду, що виконуються, менше означає краще;
- розмір — розмір мережі в мегабайтах, дозволяє оцінити, наскільки можливе розміщення мережі на пристроях з обмеженими потужностями, менше означає краще.

Метрика Mean Average Precision визначається як:

$$mAP = \frac{1}{k} \sum_i^k AP_i,$$

де k вказує на кількість класів, а AP_i на середню точність по заданому класу.

3.4.2 Середовище розробки, гіперпараметри і оптимізатор

Розробка здійснювалась на локальному комп'ютері з процесором Intel Core i5-13600K, 32 ГБ оперативної пам'яті та графічним адаптером NVIDIA A4000 з 16 ГБ відеопам'яті. Встановлена операційна система Ubuntu 20.04.6 LTS з дистрибуцією мови Python 3.10.13, PyTorch 2.2.1.

Кількість епох навчання — 250 з використанням ранньої зупинки навчання, розмір батчу — 32, оптимізатор — стохастичний градієнтний спуск зі значенням імпульсу 0.937, швидкістю навчання 0.01 і коефіцієнтом згасання ваг 0.005. Решта гіперпараметрів визначалися за допомогою бібліотеки Ray Tune.

3.4.3 Результати експериментів

Експериментально було порівняно існуючі моделі сімейства YOLO, двокроковий алгоритм Faster R-CNN, однокроковий RetinaNet та FCOS. Власна архітектура нейронної мережі була порівняна з модулем попередньої обробки із застосуванням алгоритму контраст-обмеженої гістограмної еквалізації, і без нього. Розміщення модуля C2f-G в даних експериментах — песок-частина архітектури YOLO. Доданий час для обробки кожного знімка даним алгоритмом оцінюється в ~ 2.5 мс, але залежить від пристрою, на якому виконується, тому потребуватиме додаткового уточнення. Результати експериментів для вибірки UTDAC2020 наведено в таблиці 3.2. Найкращі результати виділено жирним кеглем.

Таблиця 3.2 — Результати експериментів на вибірці UTDAC2020

<i>Методологія</i>	<i>mAp₅₀</i>	<i>mAp</i>	<i>Параметри (мільйонів)</i>	<i>FLOPs (G)</i>	<i>Розмір (МБ)</i>
Faster R-CNN	0.809	0.445	41.14	63.26	~
RetinaNet	0.804	0.439	36.17	52.62	~
FCOS	0.811	0.439	31.84	50.36	~
YOLOv8n	0.8265	0.4901	3	8.1	6
YOLOv8s	0.8471	0.5053	11.1	28.4	22
YOLOv8m	0.8492	0.5169	25.8	78.7	51
YOLOv8l	0.8497	0.5171	43.6	164.8	84
Власна (YOLOv8s + C2f-G)	0.8618	0.5245	9	27	18
Власна (YOLOv8s + C2f-G + CLAHE)	0.8633	0.5268	9	27	18

Результати дослідження показують, що за точністю запропонована архітектура випереджає моделі сімейства YOLO, знаходячись по рівні за розміром з YOLOv8s, але на 1.62% випереджаючи її в точності.

Таблиця 3.3 — Результати експериментів на вибірці URPC2019

<i>Методологія</i>	<i>mAp₅₀</i>	<i>mAp</i>	<i>Параметри (мільйонів)</i>	<i>FLOPs (G)</i>	<i>Розмір (МБ)</i>
YOLOv8n	0.803	0.4732	3	8.1	6
YOLOv8s	0.8142	0.4881	11.1	28.4	22
YOLOv8m	0.8123	0.4874	25.8	78.7	51
YOLOv8l	0.8255	0.4965	43.6	164.8	84
Власна (YOLOv8s + C2f-G)	0.851	0.5178	9	27	18
Власна (YOLOv8s + C2f-G + CLAHE)	0.8521	0.5188	9	27	18

Для підтвердження результатів модель було також випробувано на вибірці URPC2019. В результаті, модель на базі запропонованої архітектури виявилась на 3.79% точнішою за метрикою mAp₅₀. Візуалізація результатів детектування представлена на рис. 3.7.

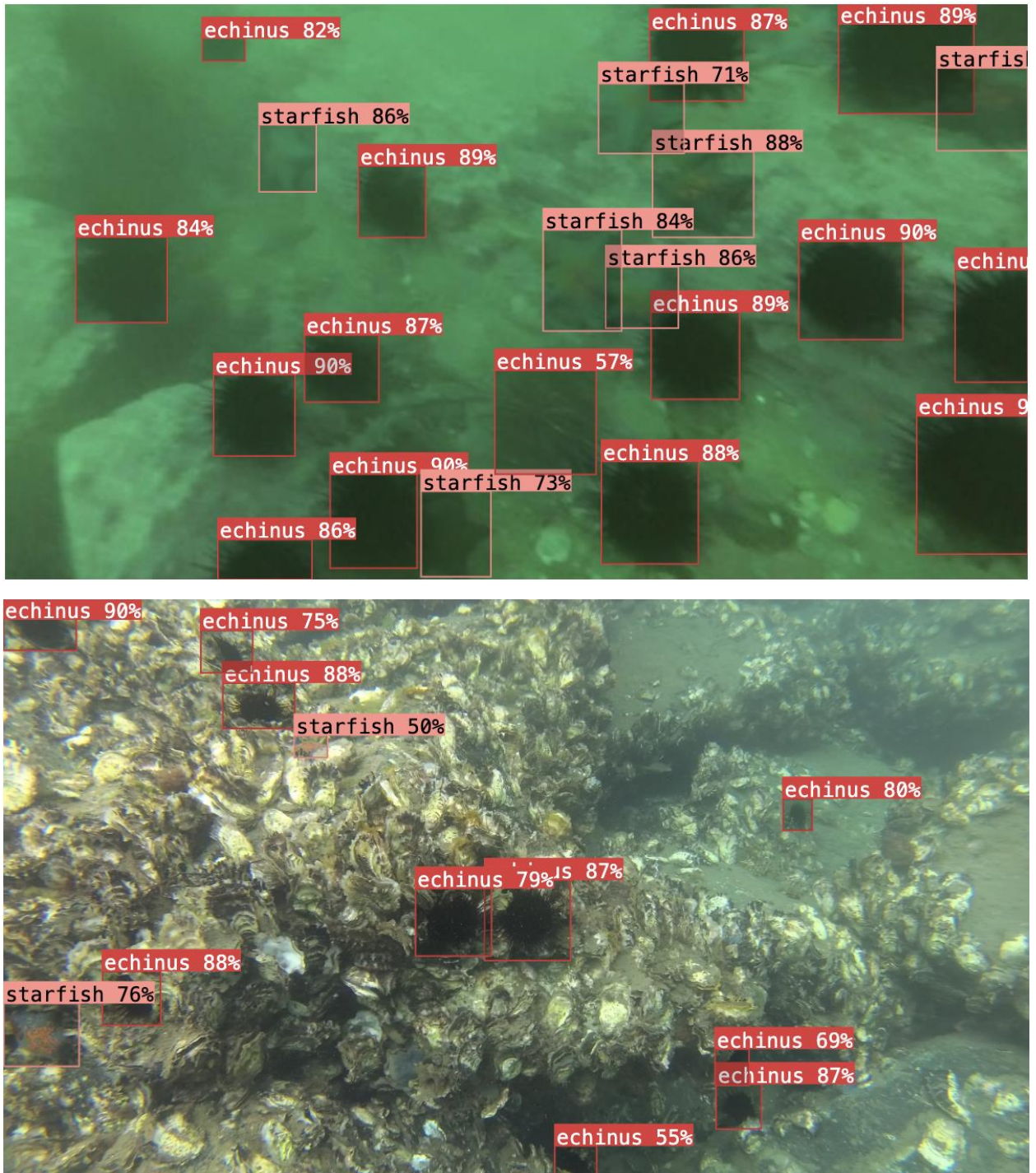


Рисунок 3.7 — Приклад детектувань об'єктів запропонованою моделлю

3.5 Структура та інтерфейс програмного забезпечення

Після побудови, навчання нейронної мережі та отримання оптимальних значень ваг, отримуємо ваги моделі у форматі .pt та експортовану модель у відкритому форматі .onnx. Модель в форматі .onnx пропонується розгорнути на окремому модулі обчислень NVIDIA Jetson Nano (рис. 3.8) з наступними характеристиками:

- чотирьох'ядерний процесор на базі архітектури ARM;
- 4 ГБ оперативної пам'яті стандарту LPDDR4;
- графічний прискорювач NVIDIA зі 128 обчислювальними ядрами CUDA;
- номінальна потужність 5 Вт.



Рисунок 3.8 — Обчислювальна платформа NVIDIA Jetson Nano

Компіляція моделі виконується в програмному пакеті Tensor RT, який виконує ряд оптимізацій для використання нейронних мереж на мобільних пристроях. Завдяки низькому енергоспоживанню в робочому режимі, така

Джерело зображення — <https://developer.nvidia.com/blog/sharpen-your-edge-ai-and-robotics-skills-with-the-nvidia-jetson-nano-developer-kit/>

обчислювальна платформа підходить для встановлення в якості окремого модуля на борт автономного підводного апарата.

Пропонується наступна структура програмного забезпечення (рис. 3.9):

- відео з камер, під'єднаних до апаратного модуля, передається на вхід програмного забезпечення OpenCV та V4L2;
- отриманий відеоряд передається на вхід нейронної мережі, яка на виході подає прогнозовані обмежувальні рамки об'єктів та мітки їхніх класів;
- відеоряд по протоколу RMTP пакетом FFmpeg і виходи з нейронної мережі передаються на клієнтську частину.

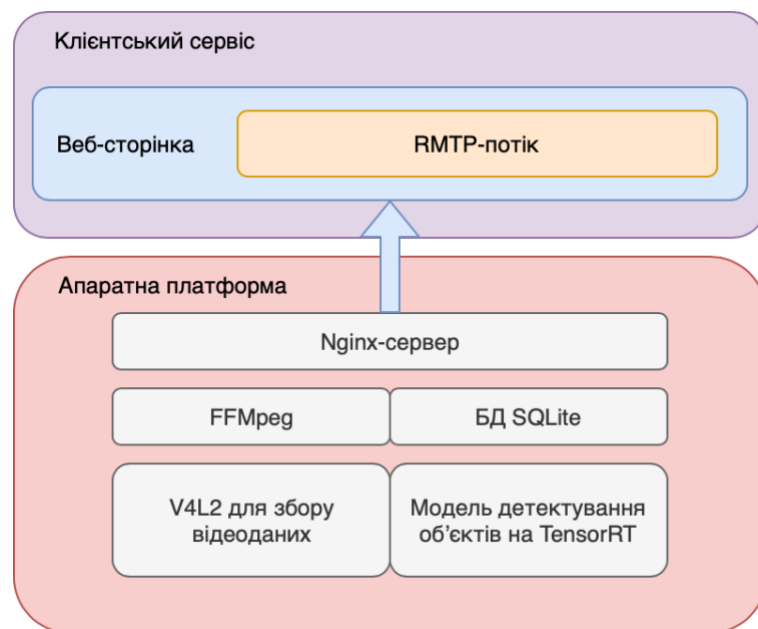


Рисунок 3.9 — Приклад архітектури програмного забезпечення

Для тестування та демонстрації можливостей нейронної мережі було побудовано веб-версію застосунку, який дозволяє обрати попередньо навчену нейронну мережу або завантажити власну у форматі .pt або .onnx, завантажити зображення для тестування і візуалізувати результати визначення об'єктів. Окрім цього, можна обрати поріг відсіювання обмежувальних рамок за ймовірністю. Інтерфейсна частина додатку побудована на базі фреймворку Streamlit на базі відкритого програмного коду (рис. 3.10).

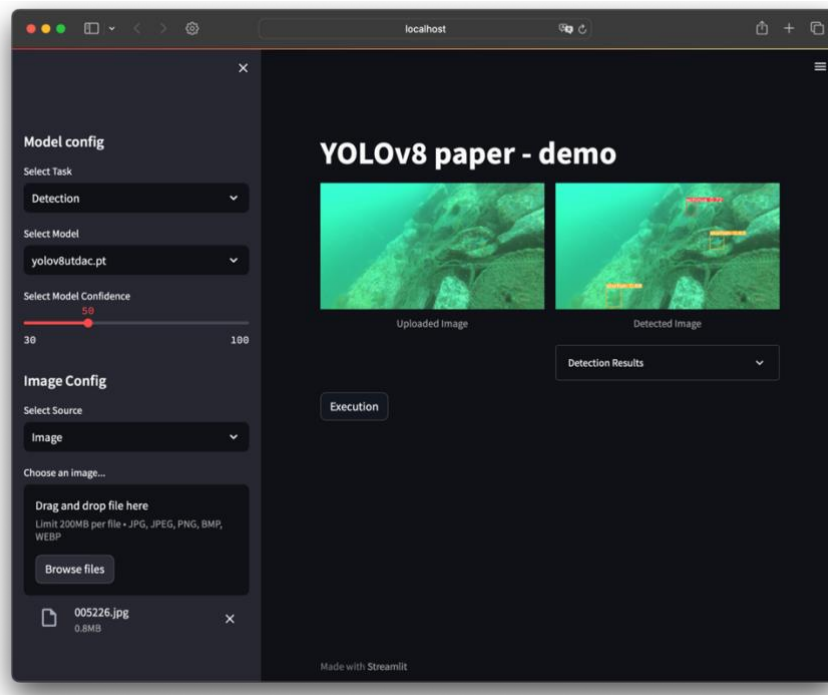


Рисунок 3.10 — Інтерфейс користувача веб-застосунку

Вибір зображення доступний через перетягування на діалогове вікно програми або через системне вікно вибору файлу (рис. 3.11):

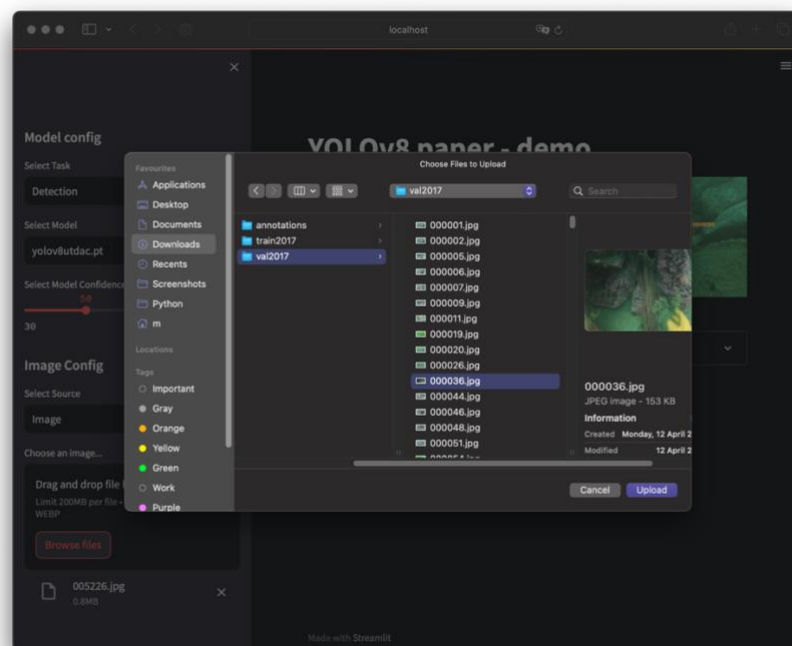


Рисунок 3.11 — Діалогове вікно вибору файлу

Висновки до розділу 3

В розділі розглянуто вимоги до програмного забезпечення інтелектуальної системи детектування об'єктів у підводному середовищі. Встановлено, що система має бути здатною ефективно виявляти і класифікувати об'єкти під водою з мінімальною кількістю хибних спрацювань, мати здатність до роботи в умовах обмеженої освітленості та з зображеннями низької якості, визначати об'єкти з частковим перекриттям або малим розміром та бути достатньо обчислювально ефективною для розміщення на малопотужних пристроях.

Програмне забезпечення реалізовано модулем попередньої обробки, нейронною мережею детектування об'єктів на основі авторських модифікацій YOLOv8 з використанням полегшеної backbone-мережі та оптимізаціями для роботи з об'єктами малих розмірів та у випадках їх перекриття й щільного розташування.

Досліджений вплив алгоритмів попередньої обробки зображення на кінцеву точність системи та час обробки одного зображення. В результаті експериментів створено модуль попередньої обробки на базі алгоритму контраст-обмеженої адаптивної гістограмної еквалізації та методів аугментації даних. Розроблено топологію нейронної мережі на базі модифікованої архітектури YOLOv8s з авторським блоком C2f-G, що забезпечує ефективне опрацювання ознак при зменшеній обчислювальній складності. Процес агрегації карт ознак вдосконалено за рахунок модуля Content Aware Reassembly of Features, який враховує вміст карти ознак при збільшенні її розміру. Результируюча точність нейронної мережі на 2-3% перевищує існуючі рішення при меншому рівні обчислювальних витрат.

Побудовану модель експортовано у відкритий формат .onnx для використання на малопотужних пристроях NVIDIA Jetson. Побудовано веб-інтерфейс застосунку для демонстрації роботи мережі.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ СИСТЕМИ ДЕТЕКТУВАННЯ

4.1 Опис ідеї проєкту

Сучасні технології обробки даних та штучного інтелекту значно розширюють можливості автоматизованих систем у різних галузях, зокрема в дослідженні морських середовищ. Однією з ключових задач є детектування підводних об'єктів, що має критичне значення для таких сфер, як океанографія, морська біологія, підводна археологія, а також забезпечення безпеки морських перевезень. Ефективна система детектування здатна автоматично аналізувати підводні зображення або відео, точно визначаючи об'єкти навіть за умов обмеженої видимості та наявності шумів.

Існуючі системи пропонують універсальний підхід до визначення об'єктів і потребують значних архітектурних допрацювань для ефективного виконання прикладних задач. Розроблена модульна система здатна вирішити проблеми відомих рішень завдяки ефективній реалізації попередньої обробки зображень для усунення оптичних перешкод та можливості роботи з цілями малих розмірів та частково перекритих об'єктів.

Розробка ідеї стартап-проєкту такої системи складається з п'яти кроків.

1. Опис ідеї проєкту, визначення сильних і слабких сторін.
2. Технологічний аудит проєкту, оцінка технологічної здійсненності.
3. Аналіз ринкових можливостей запуску стартап-проєкту.
4. Розроблення стратегії виходу проєкту на ринок.
5. Розроблення маркетингової програми для стартапу.

Метою проєкту є розробка інтелектуальної системи для детектування підводних об'єктів, яка розгортається на апаратних модулях, таких як NVIDIA Jetson або Raspberry Pi, і може бути використана на борту автономного підводного апарата. Система повинна забезпечувати високу точність та швидкість обробки даних, ефективно працюючи в реальному часі навіть за складних умов підводного середовища. Гнучкість рішення має дозволяти

адаптувати його до конкретних практичних завдань та індивідуальних вимог користувачів. Передбачається, що система буде підтримувати програмні оновлення, що дозволить вдосконалювати її функціональність з часом та відповідати новим технічним викликам. Окрім цього, система має бути оснащена механізмами захисту інформації, забезпечувати безпечне передавання даних і можливість архівування результатів для подальшого аналізу. Такий підхід уможлиблює використання системи у якості надійного інструменту для застосування у наукових, комерційних та оборонних цілях.

Короткий опис проєкту подано в табл. 4.1.

Таблиця 4.1 — Опис ідеї стартап-проєкту системи детектування підводних об'єктів

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигода для користувача</i>
Створення інтелектуальної системи детектування підводних об'єктів, яка використовує сучасні алгоритми комп'ютерного зору та глибокого навчання. Система повинна бути оптимізована для роботи в реальному часі на апаратних платформах, таких як NVIDIA Jetson, і забезпечувати високий рівень адаптивності до умов підводного середовища	Автоматизація процесу аналізу морських екосистем, моніторинг стану коралових рифів або підводних біотопів; виявлення та ідентифікація потенційно небезпечних об'єктів, таких як міни чи затонулі судна; моніторинг підводних інфраструктур (нафтогазові платформи, трубопроводи), технічне обслуговування та контроль стану об'єктів; дослідження стану забруднень водою.	Користувачі отримають надійне та інноваційне рішення для автоматизованого детектування об'єктів у підводному середовищі, яке спрощує виконання складних задач і економить ресурси. Система дозволяє працювати в реальному часі з високою точністю, адаптується до специфічних вимог користувача і забезпечує захист даних та безпечне передавання інформації..

Окрім цього, було проведено аналіз сильних, нейтральних та слабких сторін проєкту (табл. 4.2).

Таблиця 4.2 — Аналіз сильних, слабких та нейтральних характеристик стартапу

<i>Техніко-економічні характеристики ідеї</i>	<i>Оцінка</i>	<i>W (слабка сторона)</i>	<i>W (нейтральна сторона)</i>	<i>S (сильна сторона)</i>
Точність детектування об'єктів	Висока точність	+		
Швидкість роботи	В режимі реального часу	+		
Можливість роботи з наявними перешкодами	Працює з низькою якістю зображень	+		
Простота у використанні	Зручний користувацький інтерфейс		+	
Пристосованість до потреб клієнта	Можливе допрацювання під індивідуальне замовлення	+		
Конфіденційність і безпека	Шифрування імплементується за вимогами клієнта		+	

З даного аналізу можна зробити висновок про те, що запропонований проєкт є конкурентоспроможним.

4.2 Технологічний аудит проєкту

Метою технічного аудиту є ідентифікація та оцінка технологічних рішень, застосованих під час створення системи відповідно до концепції

проєкту. У рамках цього аналізу розглядається технічна реалізованість запропонованого концепту, що включає дослідження таких аспектів (табл. 4.3):

- використанні технологічні інструменти та їх відповідність основній ідеї проєкту;
- доступність обраних технологій на ринку, включаючи оцінку того, чи є вони вільними у використанні, чи потребують додаткової розробки або ліцензування;
- можливості розробників щодо доступу до цих технологій та їх практичного застосування для реалізації проєкту.

Таблиця 4.3 — Результати технологічного аудиту

<i>Частина проєкта</i>	<i>Технології реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
Модуль попередньої обробки зображень	Python-бібліотеки OpenCV, RandAugment, Albumentations	Наявні	Доступні
Модуль розпізнавання об'єктів	Python-бібліотека Ultralytics з власними модифікаціями	Наявна	Доступна
Користувацький інтерфейс	Javascript, React	Наявні	Доступні

Модуль попередньої обробки зображень реалізовано через Python-бібліотеки OpenCV, RandAugment, та Albumentations. Ці технології є широко доступними, мають відкритий вихідний код і активно підтримуються спільнотою. Наявність бібліотек на ринку гарантує їхню доступність для використання в межах проєкту без необхідності придбання ліцензій чи додаткової розробки.

Модуль розпізнавання об'єктів використовує Python-бібліотеку Ultralytics з власними модифікаціями, які адаптують базову функціональність під специфіку завдання. Ця технологія є доступною як у відкритому, так і в комерційному форматі, що дозволяє розробникам працювати з нею без

обмежень. Наявність бібліотеки на ринку та можливість її модифікації забезпечують гнучкість у реалізації проєкту.

Для створення інтерфейсу використовуються Javascript і React, які є популярними технологіями у веб-розробці. Вони також доступні у відкритому доступі, підтримуються значною розробницькою спільнотою та мають велику кількість супутніх інструментів і бібліотек. Це забезпечує можливість їхнього ефективного застосування в межах проєкту.

4.3 Аналіз ринкових можливостей запуску проєкту

Аналіз ринкових можливостей та загроз є ключовим етапом планування запуску стартап-проєкту, оскільки дозволяє оцінити потенціал впровадження продукту, адаптувати його до потреб ринку та забезпечити конкурентоспроможність. У рамках цього дослідження були визначені ринкові можливості, які сприятимуть успішній реалізації проєкту, а також ідентифіковано загрози, що можуть ускладнити його впровадження.

Ринкові можливості полягають у зростаючому попиті на автоматизовані системи для дослідження підводного середовища, збільшенні інвестицій у технології для автономних підводних апаратів (AUV) та зацікавленості у високоточних рішеннях для моніторингу та обслуговування морської інфраструктури. Крім того, широке використання відкритих бібліотек та доступність сучасних апаратних платформ, таких як NVIDIA Jetson, створюють умови для зниження витрат на розробку та впровадження системи.

Водночас ринкові загрози включають високу конкуренцію у сфері технологій для морської індустрії, можливі бар'єри у виході на міжнародний ринок через регуляторні вимоги, а також необхідність постійного оновлення продукту для відповідності швидкому розвитку технологій. Іншим важливим

викликом є забезпечення довіри користувачів до системи, що включає прозорість роботи алгоритмів, захист інформації та зручність користування.

Проведений аналіз ринкового попиту дозволив виявити основні сегменти споживачів, серед яких науково-дослідні установи, комерційні організації, що займаються обслуговуванням морських об'єктів, та оборонні відомства. На основі цих даних було складено табл. 4.4, яка систематизує ключові аспекти ринкових можливостей і загроз, що впливають на розвиток проєкту.

Таблиця 4.4 — Результати аналізу ринкових можливостей запуску

<i>№</i>	<i>Показники стану ринку</i>	<i>Характеристика</i>
1	Кількість головних гравців, од.	невідомо
2	Загальний обсяг продажів	невідомо
3	Динаміка ринку (якісна оцінка)	зростає
4	Наявність обмежень для входу	вимоги регуляторів, вимоги до навчальних даних і шифрування програмного забезпечення і даних
5	Специфічні вимоги до стандартизації та сертифікації	присутні
6	Середня норма рентабельності в галузі або по ринку, %	35

Через специфіку галузі, визначити точну кількість гравців на ринку є складним завданням, оскільки багато компаній працюють у вузьких нішах, які не завжди публічно висвітлюються. Сюди входять як великі корпорації, що розробляють комплексні системи для дослідження океанів та забезпечення безпеки, так і менші стартапи, які спеціалізуються на окремих рішеннях, таких як обробка підводних зображень або сенсорні системи.

Загалом, проєкт є привабливим з точки зору реалізації і виходу на ринок. Також було уточнено групи клієнтів та специфіка їх потреб. Результати аналізу представлено в табл. 4.5.

Таблиця 4.5 — Результати аналізу ринкових можливостей запуску

<i>№</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1	Моніторинг підводних екосистем	Науково-дослідні установи, екологічні організації	Висока зацікавленість у деталізації даних і можливості проведення довготривалих досліджень	Висока точність, можливість роботи у складних умовах, підтримка архівування даних для подальшого аналізу
2	Контроль стану підводних інфраструктур	Комерційні підприємства (нафтогазова галузь, морські порти, енергетика)	Фокус на оптимізації витрат та швидкості обробки даних	Надійність, інтеграція з існуючими системами, можливість швидкої діагностики стану інфраструктури
3	Забезпечення морської безпеки	Оборонні відомства, державні служби з безпеки	Орієнтованість на конфіденційність і захист даних	Можливість роботи у реальному часі, захист інформації, стійкість до зовнішніх загроз
4	Дослідження підводних ресурсів	Приватні компанії (видобування корисних копалин, пошук затонулих об'єктів)	Прагнення отримати максимальний комерційний ефект із мінімальними інвестиціями	Гнучкість у налаштуванні для специфічних задач, підтримка високої роздільної здатності зображень

Результати аналізу послугували джерелом для таблиці факторів, що сприяють (табл. 4.6) та перешкоджають (табл. 4.7) ринковому впровадженню запропонованого стартап-проекта.

Таблиця 4.6 — Фактори загроз

<i>№</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1	Конкуренція	Вихід на ринок продуктів з кращими показниками метрик	Вдосконалення продукту, випуск оновлень
2	Зміна потреб користувачів	Поява нових прикладних задач, які потребуватимуть нового рішення	Випуск оновлень, донавання мережі, вдосконалення інтерфейсу

Таблиця 4.7 — Фактори можливостей

<i>№</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1	Точність та швидкість виявлення об'єктів	Мережа забезпечує високу точність в умовах обмеженої видимості	Залучення нових користувачів
2	Розширення	Розширення функцій для виконання прикладних задач, які потребуватимуть нового рішення	Додавання нових функцій

Висока конкуренція та можливість зміни потреб користувачів є ключовими викликами. Для їх подолання компанія має вдосконалювати продукт, забезпечувати його актуальність через регулярні оновлення та адаптацію до нових задач.

Також виділено перспективні можливості, такі як висока точність і швидкість системи, що можуть привабити нових користувачів, а також потенціал розширення функціоналу. Їх реалізація сприятиме зростанню клієнтської бази та підвищенню конкурентоспроможності продукту.

В табл. 4.8 наведено ступеневий аналіз конкуренції на ринку.

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
Тип конкуренції: олігополія	На ринку присутні декілька подібних систем для підводних досліджень	Розробка унікальних функцій, орієнтованих на підвищення точності та швидкості роботи; акцент на перевагах продукту
За рівнем конкурентної боротьби: міжнародний	Системи для підводного обстеження пропонуються розробниками з усього світу	Орієнтація на глобальні стандарти, розширення можливостей для інтеграції системи в міжнародні екосистеми
За галузевою ознакою: міжгалузева	Продукт може бути використаний у науці, енергетиці, екології, обороні	Розробка персоналізованих рішень залежно від задач користувача
Конкуренція за видами товарів: товарно-родова	Конкуренція між різними системами, що виконують функції виявлення та класифікації	Розширення функціональності, постійні оновлення системи, акцент на обчислювальній ефективності
За характером конкурентних переваг: нецінова	Основна конкуренція базується на точності, швидкості та адаптивності системи	Впровадження високотехнологічних рішень, які забезпечують перевагу над конкурентами
За інтенсивністю: не марочна	На ринку немає відомих брендів у даній сфері	Зосередження на якості продукту, без необхідності великих інвестицій у створення бренду

З врахуванням наведених вище даних аналізу, обґрунтуємо фактори конкурентоспроможності (табл. 4.9).

Таблиця 4.9 — Ступеневий аналіз конкуренції на ринку

<i>№</i>	<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>
1	Точність	Точність ~90%
2	Швидкість	Швидке детектування і класифікація
3	Собівартість	Доступна для розміщення на малопотужних пристроях, що не відрізняються високою ціною
4	Захищеність	Є можливість адаптувати систему до персональних запитів щодо захисту інформації
5	Якість	Працює в несприятливих умовах та при обмежувальних обчислювальних ресурсах

На основі цих даних проведемо аналіз сильних та слабких сторін запропонованого рішення (табл. 4.10).

Таблиця 4.10 — Аналіз сильних та слабких сторін стартап-проекту

<i>№</i>	<i>Фактор конкуренто-спроможності</i>	<i>Бали (1-20)</i>	<i>Рейтинг товарів-конкурентів (від -3 до +3)</i>
1	Точність	18	-1
2	Швидкість	19	0
3	Собівартість	20	-2
4	Захищеність	16	+2
5	Якість	18	0

Ці дані покладено в основу матриці SWOT-аналізу (табл. 4.11), який враховує чотири фактори: сильні сторони (англ. — strengths), слабкі сторони (англ. — weaknesses), можливості (англ. — opportunities) та загрози (англ. — threats).

Таблиця 4.11 — SWOT-аналіз стартапу

<i>Сильні сторони:</i> висока точність і швидкість роботи системи, енергоефективність і оптимізація, гнучкість системи, можливість адаптації до різних задач	<i>Слабкі сторони:</i> відсутність сформованої бази постійних клієнтів, обмежений бюджет для маркетингового просування, невідомість бренду на міжнародному ринку
<i>Можливості:</i> використання системи в різних галузях, розширення функціоналу для нових прикладних задач, підтримка програмних оновлень і масштабування, можливість інтеграції з існуючими підводними апаратами	<i>Загрози:</i> недовіра користувачів до нових технологій, поява більш розвинених конкурентів, різке зниження цін конкурентами, потенційні обмеження у сфері захисту інформації

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки орієнтовний час їх ринкової реалізації з огляду на загрозу появи конкурентів на ринку (табл. 4.12).

Таблиця 4.12 — Альтернативи ринкового впровадження стартапу

№	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
1	Вихід в якості програмного продукту	80%	6 місяців
2	Вихід в якості готового апаратного модуля	35%	18 місяців

Виходячи з аналізу альтернатив ринкового впровадження стартап-проєкту, можна зробити висновок, що найбільш ймовірним та швидким шляхом для реалізації є вихід на ринок у вигляді програмного продукту. Водночас, альтернатива виходу на ринок у вигляді готового апаратного модуля потребує більше часу для реалізації, але може бути цікавою в довгостроковій перспективі для побудови власної екосистеми продуктів та більшої захищеності від конкурентів.

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії починається з вибору груп потенційних користувачів (табл. 4.13).

Таблиця 4.13 — Вибір цільових груп потенційних споживачів

<i>№</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів в сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1	Компанії, що розробляють і обслуговують підводні апарати та техніку	висока	55%	висока	середня
2	Науково-дослідні організації та університети, які займаються підводними дослідженнями	висока	65%	середня	середня
3	Компанії, що займаються обслуговуванням і розробкою морських об'єктів та інфраструктури	середня	70%	середня	середня

Визначимо базову стратегію розвитку при орієнтуванні на обрані групи користувачів, а саме: компанії, що розробляють і обслуговують підводні апарати та техніку та компанії, що займаються обслуговуванням і розробкою морських об'єктів та інфраструктури. Для роботи в обраних сегментах ринку визначено наступну базову стратегію розвитку (табл. 4.14).

Таблиця 4.14 — Визначення базової стратегії розвитку

<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку</i>
Вихід в якості програмного продукту	Диференційований маркетинг	Ефективність системи і можливість роботи за несприятливих умов	Стратегія диференціації

Також визначимо базову стратегію конкурентної поведінки (табл. 4.15) і стратегію позиціонування (табл. 4.16).

Таблиця 4.15 — Визначення базової стратегії розвитку

<i>Чи є проект унікальним на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки</i>
Ні	Шукати нових і забирати існуючих	Так, подібну структуру системи та поєднання тих же технологій	Стратегія наслідування лідера

Таблиця 4.16 — Визначення базової стратегії розвитку

<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції проекту</i>	<i>Вибір асоціацій, які мають сформулювати комплексну позицію проекту</i>
Швидкість, точність, зручність використання	Стратегія диференціації	Точність, швидкість, якість, захищеність, доступність	Ефективність системи і можливість роботи за несприятливих умов

Висновки до розділу 4

В межах цього розділу було проведено детальний аналіз ринкових можливостей для запуску стартап-проекту з розробки системи детектування підводних об'єктів. Проаналізовано сильні та слабкі сторони проекту, проведено SWOT-аналіз, а також розглянуто фактори, що впливають на конкурентне середовище та ринкові загрози. Визначено основні цільові групи клієнтів, серед яких компанії, що займаються розробкою та обслуговуванням підводних апаратів, науково-дослідні організації, а також компанії, що працюють в морській інфраструктурі та безпеці.

За результатами аналізу було встановлено, що ринок для даної системи має значний потенціал та попит на подібні технології зростає. Однак для успішної реалізації проекту необхідно враховувати високий рівень конкуренції, потребу у потужних обчислювальних ресурсах і специфічних даних для роботи з підводними апаратами. Стратегія виходу на ринок передбачає використання стратегії диференціації продукту, спрямованої на підвищення точності та швидкості роботи системи, що буде основним конкурентним перевагою.

З огляду на результати SWOT-аналізу та ринкові можливості, проект має привабливий потенціал для успіху на ринку. Рекомендовано розглянути варіанти виходу на ринок як програмного продукту або готового апаратного модуля, в залежності від наявних ресурсів і умов конкуренції. Враховуючи існуючі можливості та загрози, подальша імплементація проекту є доцільною, з акцентом на забезпечення високої конкурентоспроможності та готовність до швидкого реагування на зміни в ринковому середовищі.

ВИСНОВКИ

У результаті виконаних досліджень було розглянуто та детально проаналізовано важливі аспекти створення системи детектування об'єктів для автономних підводних апаратів. Задача виявлення та класифікації об'єктів у підводному середовищі є надзвичайно складною через специфічні умови, такі як низька видимість, розмитість зображень та часткове перекриття об'єктів. Для вирішення цієї проблеми було обрано архітектуру YOLO, яка демонструє високі результати у задачах детектування об'єктів завдяки своїй швидкості та точності.

В процесі роботи було здійснено огляд сучасних методів детектування об'єктів, серед яких методи на базі класичного машинного навчання, а також сучасні методи на основі глибокого навчання. Особлива увага була приділена методам детектування на базі архітектури YOLO, що є оптимальним вибором для задач, що потребують високої швидкості обробки. У рамках подальшого дослідження були розглянуті можливості вдосконалення цих моделей, зокрема через оптимізацію нейронних мереж, що знижує їх обчислювальні витрати.

Дослідження показало, що для створення інтелектуальної системи детектування об'єктів у підводному середовищі необхідно враховувати багато факторів, зокрема ефективність алгоритмів попередньої обробки зображень, що покращують точність і зменшують час обробки.

В результаті експериментів створено модуль попередньої обробки на базі алгоритму контраст-обмеженої адаптивної гістограмної еквалізації та методів аугментації даних. Розроблено топологію нейронної мережі на базі модифікованої архітектури YOLOv8s з авторським блоком C2f-G, що забезпечує ефективне опрацювання ознак при зменшеній обчислювальній складності. Процес агрегації карт ознак вдосконалено за рахунок модуля Content Aware Reassembly of Features, який враховує вміст карти ознак при

збільшенні її розміру. Результуюча точність нейронної мережі становить 86,33% при валідації на вибірці підводних зображень Underwater Target Detection Algorithm Competition 2020, яка складається зі знімків біологічних організмів у океанічному середовищі. Отримані результати на 2-3% перевищують найкращі існуючі рішення при меншому рівні обчислювальних витрат.

Модель була експортована у формат .onnx для використання на малопотужних пристроях, таких як NVIDIA Jetson, і продемонстрована через веб-інтерфейс.

Відповідно до аналізу ринкових можливостей для запуску стартапу з розробки системи детектування підводних об'єктів було встановлено, що попит на подібні технології значно зростає. При цьому важливим фактором для успіху проекту є конкурентоспроможність продукту, яка може бути досягнута через підвищення точності та швидкості роботи системи. Прогнозується, що проект має великий потенціал на ринку, якщо будуть враховані всі технічні та бізнес-аспекти, зокрема необхідність у потужних обчислювальних ресурсах для постійного донавчання моделей.

Встановлено, що реалізація системи детектування підводних об'єктів є доцільною та перспективною. Подальша імплементація проекту вимагає акценту на високій конкурентоспроможності, адаптації до змін на ринку та гнучкості в розробці продукту з урахуванням новітніх технологій і вимог користувачів.

Основні положення роботи викладено в роботі В. Синєглазова, М. Савченка і В. Литвиненка "Efficient small and overlapping target detection in underwater images" та представлено на 4th Int. Workshop IT-professionals Artif. Intell. (ProfIT AI 2024) у вересні 2024 року [44].

ПЕРЕЛІК ПОСИЛАНЬ

- [1] Z. Zou, K. Chen, Z. Shi, Y. Guo та J. Ye, “Object Detection in 20 Years: A Survey”, *Proc. IEEE*, с. 1–20, 2023. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/jproc.2023.3238524>
- [2] P. Viola та M. Jones, “Rapid object detection using a boosted cascade of simple features”, у 2001 *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.. CVPR 2001*, Kauai, HI, USA. *IEEE Comput. Soc*, б. д. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr.2001.990517>
- [3] N. Dalal та B. Triggs, “Histograms of Oriented Gradients for Human Detection”, у 2005 *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR'05)*, San Diego, CA, USA. *IEEE*, б. д. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr.2005.177>
- [4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester та D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models”, *IEEE Trans. Pattern Anal. Mach. Intell.*, т. 32, № 9, с. 1627–1645, верес. 2010. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/tpami.2009.167>
- [5] Y. Sun, Z. Sun та W. Chen, “The evolution of object detection methods”, *Eng. Appl. Artif. Intell.*, т. 133, с. 108458, лип. 2024. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1016/j.engappai.2024.108458>
- [6] A. Y. Virasova, D. I. Klimov, O. E. Khromov, I. R. Gubaidullin та V. V. Oreshko, “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Radioengineering*, с. 115–126, 2021. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.18127/j00338486-202109-11>
- [7] K. He, X. Zhang, S. Ren та J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”, *IEEE Trans. Pattern Anal.*

- Mach. Intell., т. 37, № 9, с. 1904–1916, верес. 2015. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/tpami.2015.2389824>
- [8] R. Girshick, “Fast R-CNN”, у 2015 IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, 7–13 груд. 2015. IEEE, 2015. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/iccv.2015.169>
- [9] S. Ren, K. He, R. Girshick та J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Trans. Pattern Anal. Mach. Intell., т. 39, № 6, с. 1137–1149, черв. 2017. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/tpami.2016.2577031>
- [10] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan та S. Belongie, “Feature Pyramid Networks for Object Detection”, у 2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, 21–26 лип. 2017. IEEE, 2017. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr.2017.106>
- [11] J. Redmon, S. Divvala, R. Girshick та A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, у 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 27–30 черв. 2016. IEEE, 2016. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr.2016.91>
- [12] W. Liu та ін., “SSD: Single Shot MultiBox Detector”, у Computer Vision – ECCV 2016. Cham: Springer Int. Publishing, 2016, с. 21–37. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: https://doi.org/10.1007/978-3-319-46448-0_2
- [13] A. Vaswani та ін. “Attention Is All You Need”. arXiv.org. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://arxiv.org/abs/1706.03762>
- [14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov та S. Zagoruyko, “End-to-End Object Detection with Transformers”, у Computer Vision – ECCV 2020. Cham: Springer Int. Publishing, 2020, с. 213–229. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: https://doi.org/10.1007/978-3-030-58452-8_13

- [15] M. Zgurovsky, V. Sineglazov та E. Chumachenko, *Artificial Intelligence Systems Based on Hybrid Neural Networks*. Cham: Springer Int. Publishing, 2021. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1007/978-3-030-48453-8>
- [16] V. Sineglazov та M. Savchenko, “Comprehensive Framework for Underwater Object Detection Based on Improved YOLOv8”, *Electron. Control Syst.*, т. 1, № 79, с. 9–15, берез. 2024. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.18372/1990-5548.79.18429>
- [17] S. V.M. та С. О.І., “Structural-parametric synthesis of deep learning neural networks”, *Artif. Intell.*, т. 25, № 4, с. 42–51, груд. 2020. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.15407/jai2020.04.042>
- [18] “Histogram Equalization”, у *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2009, с. 707. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: https://doi.org/10.1007/978-0-387-73003-5_629
- [19] C. W. J. Kurak, “Adaptive Histogram Equalization, a Parallel Implementation”, *UNF Digit. Commons*, 1990. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <http://digitalcommons.unf.edu/etd/260>
- [20] S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas та K. E. Muller, “Contrast-limited adaptive histogram equalization: speed and effectiveness”, у [1990] *First Conf. Visualization Biomed. Comput.*, Atlanta, GA, USA. IEEE Comput. Soc. Press, б. д. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/vbc.1990.109340>
- [21] Kaiming He, Jian Sun, and Xiaoou Tang, “Single Image Haze Removal Using Dark Channel Prior”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/tpami.2010.168>
- [22] O. Kupyh, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, “DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks”, in *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*,

- Salt Lake City, UT, Jun. 18–23, 2018. IEEE, 2018. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr.2018.00854>
- [23] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, 2018. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <http://arxiv.org/abs/1804.02767v1>.
- [24] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://arxiv.org/abs/2004.10934>.
- [25] Ultralytics, GitHub - ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite, 2020. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://github.com/ultralytics/yolov5>.
- [26] Ultralytics, Ultralytics YOLO Docs, 2023. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://docs.ultralytics.com>.
- [27] C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao, YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, 2024. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://arxiv.org/abs/2402.13616>.
- [28] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-Time End-to-End Object Detection, 2024. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://arxiv.org/abs/2405.14458>.
- [29] N. Bodla, B. Singh, R. Chellappa та L. S. Davis, “Soft-NMS — Improving Object Detection with One Line of Code”, у 2017 IEEE Int. Conf. Comput. Vis. (ICCV), Venice, 22–29 жовт. 2017. IEEE, 2017. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/iccv.2017.593>
- [30] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye та D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”, Proc. AAAI Conf. Artif. Intell., т. 34, № 07, с. 12993–13000, квіт. 2020. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1609/aaai.v34i07.6999>
- [31] Z. Zheng та ін. “Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation”. arXiv.org. Дата

- звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://arxiv.org/abs/2005.03572>
- [32] Н. Zhang, Y. Wang, F. Dayoub та N. Sunderhauf, “VarifocalNet: An IoU-aware Dense Object Detector”, у 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, 20–25 черв. 2021. IEEE, 2021. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr46437.2021.00841>
- [33] “Gradient Descent”, у Algorithms for Convex Optimization. Cambridge Univ. Press, 2021, с. 84–107. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1017/9781108699211.008>
- [34] В. Hassibi, D. G. Stork та G. J. Wolff, “Optimal Brain Surgeon and general network pruning”, у IEEE Int. Conf. Neural Netw., San Francisco, CA, USA. IEEE, б. д. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/icnn.1993.298572>
- [35] J. Chen та ін., “Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks”, у 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Vancouver, BC, Canada, 17–24 черв. 2023. IEEE, 2023. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr52729.2023.01157>
- [36] К. Han, Y. Wang, Q. Tian, J. Guo, C. Xu та С. Xu, “GhostNet: More Features From Cheap Operations”, у 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, 13–19 черв. 2020. IEEE, 2020. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/cvpr42600.2020.00165>
- [37] К. Liu, L. Peng та S. Tang, “Underwater Object Detection Using TC-YOLO with Attention Mechanisms”, Sensors, т. 23, № 5, с. 2567, лют. 2023. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.3390/s23052567>
- [38] М. Zhang, Z. Wang, W. Song, D. Zhao та Н. Zhao, “Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network”, Appl.

- Sci., т. 14, № 3, с. 1095, січ. 2024. Дата звернення: 10 груд. 2024. [Онлайн].
Доступно: <https://doi.org/10.3390/app14031095>
- [39] M. Zhang, S. Xu, W. Song, Q. He та Q. Wei, “Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion”, *Remote Sens.*, т. 13, № 22, с. 4706, листоп. 2021. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.3390/rs13224706>
- [40] S. Zhao, J. Zheng, S. Sun та L. Zhang, “An Improved YOLO Algorithm for Fast and Accurate Underwater Object Detection”, *Symmetry*, т. 14, № 8, с. 1669, серп. 2022. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.3390/sym14081669>
- [41] X. Chen, M. Yuan, Q. Yang, H. Yao та H. Wang, “Underwater-YCC: Underwater Target Detection Optimization Algorithm Based on YOLOv7”, *J. Mar. Sci. Eng.*, т. 11, № 5, с. 995, трав. 2023. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.3390/jmse11050995>
- [42] Y. Li, X. Bai та C. Xia, “An Improved YOLOV5 Based on Triplet Attention and Prediction Head Optimization for Marine Organism Detection on Underwater Mobile Platforms”, *J. Mar. Sci. Eng.*, т. 10, № 9, с. 1230, верес. 2022. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.3390/jmse10091230>
- [43] Z. Sun та Y. Lv, “Underwater attached organisms intelligent detection based on an enhanced YOLO”, у 2022 IEEE Int. Conf. Elect. Eng., Big Data Algorithms (EEBDA), Changchun, China, 25–27 лют. 2022. IEEE, 2022. Дата звернення: 10 груд. 2024. [Онлайн]. Доступно: <https://doi.org/10.1109/eebda53927.2022.9744761>
- [44] V. Sineglazov, M. Savchenko, and V. Lytvynenko, "Efficient small and overlapping target detection in underwater images," *Proc. 4th Int. Workshop IT-professionals Artif. Intell. (ProfIT AI 2024)*, Sept. 2024, pp. 16–29, *CEUR Workshop Proc.*, vol. 3777. [Онлайн]. Доступно: <https://ceur-ws.org/Vol-3777/paper2.pdf>.

ДОДАТОК А ЛІСТИНГ КЛЮЧОВИХ СКЛАДНИКІВ ПРОГРАМИ

Preprocessing

```
import cv2
import numpy as np

def apply_clahe_to_color_image(image: np.ndarray, clip_limit: float = 2.0) -> np.ndarray:
    """
    Applies CLAHE (Contrast Limited Adaptive Histogram Equalization)
    to a color image.

    Parameters:
        image (np.ndarray): Input color image as a NumPy array (BGR
        format).
        clip_limit (float): Threshold for contrast limiting. Default
        is 2.0.

    Returns:
        np.ndarray: Image with CLAHE applied in RGB format.
    """
    # Convert the image to the LAB color space
    lab_image = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)

    # Split the LAB image into L, A, and B channels
    l_channel, a_channel, b_channel = cv2.split(lab_image)

    # Create a CLAHE object
    clahe = cv2.createCLAHE(clipLimit=clip_limit, tileGridSize=(8, 8))

    # Apply CLAHE to the L channel
    l_channel_clahe = clahe.apply(l_channel)

    # Merge the CLAHE-processed L channel back with the original A and
    B channels
    lab_image_clahe = cv2.merge((l_channel_clahe, a_channel,
    b_channel))

    # Convert the LAB image back to the RGB color space
    rgb_image_clahe = cv2.cvtColor(lab_image_clahe, cv2.COLOR_LAB2RGB)

    return rgb_image_clahe
```

Ghost blocks

```
class GhostConv(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size=1,
    stride=1, activation=True):
        super(GhostConv, self).__init__()
        self.primary_conv = nn.Conv2d(in_channels, out_channels // 2,
        kernel_size, stride, kernel_size // 2, bias=False)
        self.cheap_operation = nn.Conv2d(out_channels // 2,
        out_channels // 2, 3, 1, 1, groups=out_channels // 2, bias=False)
        self.activation = nn.ReLU(inplace=True) if activation else
        nn.Identity()

    def forward(self, x):
        primary = self.primary_conv(x)
```

```

        cheap = self.cheap_operation(primary)
        return self.activation(torch.cat([primary, cheap], dim=1))

class GhostBottleneck(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size,
stride):
        super(GhostBottleneck, self).__init__()
        self.stride = stride

        self.conv1 = GhostConv(in_channels, out_channels,
kernel_size=1, activation=True)
        self.depthwise_conv = nn.Conv2d(out_channels, out_channels,
kernel_size, stride, kernel_size // 2, groups=out_channels,
bias=False)
        self.conv2 = GhostConv(out_channels, out_channels,
kernel_size=1, activation=False)

        self.shortcut = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=1,
stride=stride, bias=False),
            nn.BatchNorm2d(out_channels)
        ) if stride != 1 or in_channels != out_channels else
nn.Identity()

    def forward(self, x):
        shortcut = self.shortcut(x)
        x = self.conv1(x)
        x = self.depthwise_conv(x)
        x = self.conv2(x)
        return x + shortcut

class C2fG(nn.Module):
    """Faster Implementation of CSP Bottleneck with Ghost
Convolutions."""
    def __init__(self, c1, c2, n=1, shortcut=False, g=1, e=0.5):
        """Initialize C2fGhost layer with arguments ch_in, ch_out,
number, shortcut, groups, expansion."""
        super(C2fG, self).__init__()
        self.c = int(c2 * e) # hidden channels
        self.cv1 = GhostConv(c1, 2 * self.c, 1, 1)
        self.cv2 = GhostConv((2 + n) * self.c, c2, 1) # Output ghost
convolution
        self.m = nn.ModuleList(GhostBottleneck(self.c, self.c,
kernel_size=3, stride=1) for _ in range(n))

    def forward(self, x):
        """Forward pass through C2fGhost layer."""
        y = list(self.cv1(x).chunk(2, 1)) # Split into two parts
        y.extend(m(y[-1]) for m in self.m) # Apply GhostBottlenecks
        return self.cv2(torch.cat(y, 1)) # Concatenate and apply
final GhostConv

    def forward_split(self, x):
        """Forward pass using split() instead of chunk()."""
        y = list(self.cv1(x).split((self.c, self.c), 1))
        y.extend(m(y[-1]) for m in self.m)
        return self.cv2(torch.cat(y, 1))

# Soft-NMS

import numpy as np

def soft_nms(detections, sigma=0.5, Nt=0.3, threshold=0.001,
method=2):

```

```

"""
Applies Soft-NMS to the detected bounding boxes.

Args:
    detections (numpy.ndarray): Array of detections [x1, y1, x2,
y2, score].
    sigma (float): Variance for the Gaussian method.
    Nt (float): IoU threshold for suppressing overlapping boxes.
    threshold (float): Score threshold for filtering detections.
    method (int): 1 for linear, 2 for Gaussian, 0 for original
NMS.

Returns:
    numpy.ndarray: Filtered detections after Soft-NMS.
"""
N = detections.shape[0]
for i in range(N):
    max_score_idx = i + np.argmax(detections[i:, 4])
    detections[[i, max_score_idx]] = detections[[max_score_idx,
i]]
    box_i = detections[i, :4]

    for j in range(i + 1, N):
        box_j = detections[j, :4]
        iou = compute_iou(box_i, box_j)

        if method == 1: # Linear
            weight = 1 - iou if iou > Nt else 1
        elif method == 2: # Gaussian
            weight = np.exp(-(iou ** 2) / sigma)
        else: # Original NMS
            weight = 0 if iou > Nt else 1

        detections[j, 4] *= weight
        if detections[j, 4] < threshold:
            detections[j, 4] = 0

detections = detections[detections[:, 4] > threshold]
return detections

def compute_iou(box1, box2):
    """
    Computes Intersection over Union (IoU) between two boxes.

    Args:
        box1 (numpy.ndarray): Box 1 [x1, y1, x2, y2].
        box2 (numpy.ndarray): Box 2 [x1, y1, x2, y2].

    Returns:
        float: IoU value.
    """
    inter_x1 = max(box1[0], box2[0])
    inter_y1 = max(box1[1], box2[1])
    inter_x2 = min(box1[2], box2[2])
    inter_y2 = min(box1[3], box2[3])

    inter_area = max(0, inter_x2 - inter_x1 + 1) * max(0, inter_y2 -
inter_y1 + 1)

    box1_area = (box1[2] - box1[0] + 1) * (box1[3] - box1[1] + 1)
    box2_area = (box2[2] - box2[0] + 1) * (box2[3] - box2[1] + 1)

    iou = inter_area / (box1_area + box2_area - inter_area)
    return iou

```

```

class CARAFE(nn.Module):
    """CARAFE: Content-Aware ReAssembly of FEatures
    https://arxiv.org/pdf/1905.02188.pdf"""
    def __init__(self, c1, c2, kernel_size=3, up_factor=2):
        super().__init__()
        self.kernel_size = kernel_size
        self.up_factor = up_factor
        self.down = nn.Conv2d(c1, c1 // 4, 1)
        self.encoder = nn.Conv2d(
            c1 // 4,
            self.up_factor**2 * self.kernel_size**2,
            self.kernel_size,
            1,
            self.kernel_size // 2,
        )
        self.out = nn.Conv2d(c1, c2, 1)

    def forward(self, x):
        N, C, H, W = x.size()

        kernel_tensor = self.down(x) # (N, Cm, H, W)
        kernel_tensor = self.encoder(kernel_tensor)
        kernel_tensor = F.pixel_shuffle(kernel_tensor, self.up_factor)
        kernel_tensor = F.softmax(kernel_tensor, dim=1)
        kernel_tensor = kernel_tensor.unfold(2, self.up_factor,
step=self.up_factor)
        kernel_tensor = kernel_tensor.unfold(3, self.up_factor,
step=self.up_factor)
        kernel_tensor = kernel_tensor.reshape(
            N, self.kernel_size**2, H, W, self.up_factor**2
        )
        kernel_tensor = kernel_tensor.permute(0, 2, 3, 1, 4)

        x = F.pad(
            x,
            pad=(
                self.kernel_size // 2,
                self.kernel_size // 2,
                self.kernel_size // 2,
                self.kernel_size // 2,
            ),
            mode="constant",
            value=0,
        )
        x = x.unfold(2, self.kernel_size, step=1)
        x = x.unfold(3, self.kernel_size, step=1)
        x = x.reshape(N, C, H, W, -1)
        x = x.permute(0, 2, 3, 1, 4)

        out_tensor = torch.matmul(x, kernel_tensor)
        out_tensor = out_tensor.reshape(N, H, W, -1)
        out_tensor = out_tensor.permute(0, 3, 1, 2)
        out_tensor = F.pixel_shuffle(out_tensor, self.up_factor)
        out_tensor = self.out(out_tensor)
        return out_tensor

```