

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Система моніторингу об'єктів морської акваторії
з використанням методів Штучного Інтелекту»**

зі спеціальності 122 Комп'ютерні науки

освітньо-професійної програми Штучний інтелект

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Артем БЄЛЯВЦЕВ

(підпис)

Виконав: здобувач вищої освіти групи ШІД-42

Артем БЄЛЯВЦЕВ

Керівник *к.т.н., доцент* Звенігородський Олександр

Рецензент

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Штучного інтелекту
Ступінь вищої освіти Бакалавр
Спеціальність 122 Комп'ютерні науки
Освітньо-професійна програма штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедри ШІ

Ольга ЗІНЧЕНКО

“ ” 2025 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Белявцеву Артему Олеговичу

1. Тема кваліфікаційної роботи: «Система моніторингу об'єктів морської акваторії з використанням методів ШІ»

керівник кваліфікаційної роботи К.т.н, доц. Зненігородський О.С
затверджені наказом Державного університету інформаційно-комунікаційних технологій від 24.02.2025р № 56

2. Строк подання кваліфікаційної роботи 23.05.2025 р.

3. Вихідні дані до кваліфікаційної роботи: Вибірка зображень, що містять різні види кораблів елементи для тренування та тестування моделей глибокого навчання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз існуючих методів і підходів до розпізнавання елементів.
2. Опис процесу створення та навчання нейронної мережі
3. Процес перевірки, виправлення помилок та підвищення ефективності веб-додатку.

5. Перелік графічного матеріалу - Презентація PowerPoint.

6. Дата видачі завдання: 25.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз предметної області	15.03.2025 р.	виконано
2.	Огляд літератури з методів розпізнавання зображень і технологій	31.03.2025 р.	виконано
3.	Вибір алгоритмів та засобів візуального розпізнавання суден з використанням методів штучного інтелекту	10.04.2025 р.	виконано
4.	Розробка структури веб-застосунку та обробка тестових зображень	25.04.2025 р.	виконано
5.	Реалізація основного функціоналу з обробки зображень у реальному часі	10.05.2025 р.	виконано
6.	Створення графічного веб-інтерфейсу користувача з можливістю інтерактивного керування	13.05.2025 р.	виконано
7.	Проведення тестування, оцінка точності роботи застосунку в різних умовах	15.05.2025 р.	виконано
8.	Підготовка пояснювальної записки та доповіді до захисту	18.05.2025 р.	виконано

Здобувач вищої освіти

Артем БЄЛЯВЦЕВ

(

Керівник

кваліфікаційної роботи

Олександр ЗВЕНІГОРОДСЬКИЙ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи - 52 сторінки, 8 рисунків, 18 джерел.

Мета роботи - Підвищення ефективності існуючих систем управління в портах шляхом впровадження алгоритмів ідентифікації суден на основі візуального розпізнавання.

Об'єкт дослідження - Процеси інформаційні та технічні системи, що функціонують у морських портах.

Предмет дослідження - Алгоритми візуального розпізнавання суден з використанням методів штучного інтелекту.

Методи дослідження - опрацювання літератури за даною темою, аналіз експлуатаційної документації.

Короткий зміст - У дипломній роботі розглядається задача розробки системи моніторингу морської акваторії з використанням сучасних методів комп'ютерного зору та штучного інтелекту. У першому розділі обґрунтовано актуальність теми, сформульовано мету та завдання дослідження. У другому розділі проведено аналіз предметної області, розглянуто існуючі системи спостереження та методи візуального розпізнавання об'єктів на морі. У третьому розділі описано архітектуру запропонованої системи, реалізацію підсистем збору відеоданих, детекції та інтерфейсу користувача, а також використання моделі YOLOv11 для виявлення суден. Робота завершується висновками та напрямками подальших досліджень.

КЛЮЧОВІ СЛОВА - СИСТЕМА МОНІТОРИНГУ, ВІЗУАЛЬНЕ РОЗПІЗНАВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, КОМП'ЮТЕРНИЙ ЗІР, ІДЕНТИФІКАЦІЯ СУДЕН, YOLOV11, ВІДЕОАНАЛІТИКА, ДЕТЕКЦІЯ ОБ'ЄКТІВ, СПОСТЕРЕЖЕННЯ НА МОРІ

ЗМІСТ

1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО МОНІТОРИНГУ МОРСЬКОЇ АКВАТОРІЇ	11
1.1 Роль моніторингу морської акваторії та характеристика традиційних методів.....	11
1.2 Сучасні технології, роль штучного інтелекту та приклади їх використання.....	16
2 ОГЛЯД ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ МОРСЬКОГО МОНІТОРИНГУ.....	22
2.1 Вибір методів візуального розпізнавання об'єктів на морі.....	22
2.2 Вимоги до системи моніторингу та розпізнавання суден.....	32
2.3 Розробка моделі системи моніторингу.....	35
3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ.....	36
3.1 Архітектура системи моніторингу об'єктів морської акваторії.....	36
3.2 Підсистема збору та обробки відеоданих із камер.....	37
3.3 Вибір алгоритмів виявлення та класифікації суден на зображеннях.....	42
3.4 Інтеграція алгоритмів у єдиний програмний комплекс	45
3.5 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ.....	49
3.5.1 Опис експериментального середовища та тестових сценаріїв.....	50
3.5.2 Результати тестування точності розпізнавання та виявлення.....	52
ВИСНОВКИ.....	56
ПЕРЕЛІК ПОСИЛАНЬ	57

ВСТУП

Актуальність дослідження - У сучасному світі спостерігається стрімкий розвиток цифрових технологій, що сприяє зростанню обсягу візуальної інформації. Автоматичне розпізнавання та обробка цієї інформації є важливою задачею для багатьох галузей, таких як архітектура, будівництво, інженерія, медицина та освіта.

Методи глибокого навчання, зокрема конволюційні нейронні мережі, демонструють високу ефективність у розпізнаванні та класифікації візуальних об'єктів. Використання цих методів у веб-додатку забезпечує високу точність та надійність розпізнавання геометричних елементів на зображеннях.

Наукова новизна і практична значущість - Наукова новизна полягає в розробці та реалізації системи моніторингу суден у морській акваторії на основі глибокого навчання з інтеграцією у веб-додаток. Запропонована система дозволяє виявляти судна на вхідних зображеннях у режимі майже реального часу. Практична значущість полягає в можливості застосування розробленої системи в умовах українських портів і служб охорони узбережжя, а також як основи для створення повноцінних аналітичних платформ для морського моніторингу.

Мета дослідження - Підвищення ефективності існуючих систем управління в портах шляхом впровадження алгоритмів ідентифікації суден на основі візуального розпізнавання.

Об'єкт дослідження - Процеси інформаційні та технічні системи, що функціонують у морських портах.

Предмет дослідження - Алгоритми візуального розпізнавання суден з використанням методів штучного інтелекту.

Методи дослідження - У роботі застосовано методи аналізу предметної області, елементів системного аналізу, машинного навчання (конволюційні нейронні мережі), програмної реалізації моделей у середовищі Python.

1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО МОНІТОРИНГУ МОРСЬКОЇ АКВАТОРІЇ

1.1 Роль моніторингу морської акваторії та характеристика традиційних методів

Моніторинг морської акваторії є критично важливим для безпеки судноплавства та збереження екосистеми. Він дозволяє оперативно реагувати на небезпеки та вчасно виявляти зміни довкілля (забруднення води, масові цвітіння водоростей тощо) [1]. За словами дослідників, впровадження автоматизованих систем спостереження, підсилених штучним інтелектом, може значно зменшити витрати на моніторинг та розширити масштаб спостережень [1].

Наприклад, нові цифрові інструменти, такі як SUBSIM (Швеція), використовують машинне навчання для аналізу підводних зображень і автоматичної ідентифікації морських видів [3], що прискорює оцінку стану підводних екосистем. На відміну від традиційних методів моніторингу морських акваторій які базуються на поєднанні геоспостережень та безпосередніх спостережень. До основних підходів належать: супутниковий моніторинг (оптичні й радарні знімки), берегова та морська радіолокація, патрулювання кораблями та літаками, використання стаціонарних (прибережних чи буйових) сенсорів і система автоматичної ідентифікації суден (AIS). Кожен із цих методів має власний обсяг покриття, чутливість і обмеження, тож у практиці їх часто комбінують для підвищення ефективності. Нижче розглянемо кожен з традиційних методів, наведемо приклади застосування та порівняємо їхні переваги й недоліки.

Супутниковий моніторинг.

Супутники з оптичними датчиками і синтетичною апертурою (SAR) дозволяють спостерігати великі площі моря одночасно. Наприклад, європейська програма Copernicus (Sentinel-2) надає оптичні знімки розміром 100×100 км ($10\,000$ км²)[2] з просторовим розрізненням 10–60 м, а комерційні супутники можуть досягати розрізнення на рівні ~25 см (хоча із значно меншим полем зору <200 км²).

Радіолокаційні (SAR) супутники цінні тим, що працюють за будь-яких погодних умов і вночі: вони здатні виявляти особливості на поверхні моря за будь-якої видимості, дозволяючи своєчасно фіксувати розливи нафти та інші аномалії. У багатьох країнах супутникові дані використовують для екологічного моніторингу [1]. Наприклад, Державна екологічна інспекція України використовує сервіс NASA Worldview (EOSDIS) для аналізу стану Чорного моря - за даними супутникових знімків фахівці виявили відсутність масового цвітіння фітопланктону у прибережних районах у літні місяці [4]. Нижче приведені переваги і недоліки:

Переваги: Надзвичайно велике покриття - одним кадром можна охопити тисячі квадратних кілометрів водної поверхні (Sentinel-2: $\sim 10\,000\text{ км}^2$). Можливість багато спектрального аналізу (оптичні канали, ІЧ тощо) дозволяє ідентифікувати типи забруднень (нафта, цвітіння водоростей тощо). SAR-супутники виявляють об'єкти незалежно від освітлення і хмарності, що дає змогу виявляти розливи нафти навіть вночі. Багато знімків (зокрема Sentinel) доступні безкоштовно, що суттєво знижує витрати на моніторинг.

Недоліки: Обмежена тимчасова роздільна здатність. Типові супутники полярних орбіт охоплюють ту саму область за один проліт, тобто в найкращому разі раз на добу (часто й рідше). Оперативне реагування на раптовий інцидент може бути ускладнене. Високий рівень деталізації (метровий чи сантиметровий) значно зменшує площу кадра і зазвичай потребує купівлі платних зображень. Оптичні супутникові системи не можуть працювати за суцільної хмарності чи вночі. Багато дрібних подій залишаються непоміченими - наприклад, тонкі нафтові розливи чи невеликі човни можуть просто бути непомітні у загальному ракурсі низькороздільного знімка. Обробка великих обсягів даних вимагає значних обчислювальних ресурсів.

AIS (Automatic Identification System).

Автоматична ідентифікація суден - це радіотранспондерна система, що передає позицію, курс, швидкість і ідентифікаційні дані судна у реальному часі. AIS є обов'язковим для більшості комерційних суден за міжнародними правилами. Його перевага в тому, що він практично не залежить від погоди чи освітлення: дані AIS регулярно отримують як берегові станції, так і супутники по всьому світу. Наприклад, аналітики відзначають, що AIS можна приймати навіть із космосу, що дозволяє відстежувати рух суден у віддалених районах океану. У багатьох країнах AIS-дані використовуються для контролю суднопотоку й попередження зіткнень, а також для аналізу рибальської активності. Нижче приведені переваги і недоліки:

Переваги: AIS працює за будь-якої погоди і видимості (дощ, туман, ніч). Передавання відбувається регулярно (від кількох разів на годину до разу за хвилину), тому інформація про рух суден оновлюється майже миттєво. Дані AIS мають високу точність GPS-координат. Більшість AIS-треків публічно доступні, що робить систему практично безкоштовною для аналітики.

Недоліки: AIS легко вимикати або фальсифікувати. Оператор судна може свідомо передавати хибні дані або взагалі відключити передавач, створюючи «темні» ділянки траєкторії. Такі випадки часто трапляються в конфліктних зонах або для приховування незаконних дій. Навіть у нормальних умовах існують «сліпі зони»: у вузьких протоках чи регіонах з високою щільністю трафіку сигнали можуть накладатися. Тому AIS не забезпечує 100% виявлення всіх суден. Саме тому міжнародні програми з контролю рибальства часто комбінують AIS з іншими методами - тільки так можна «не пропустити жодного судна».

Радіолокація та патрулювання.

Патрульні кораблі, прибережні радарні станції та повітряні патрулі (вертольоти, літаки) - класичні засоби прямого контролю. Берегові радіолокатори (Х- або S-діапазону) фіксують усі великі металеві об'єкти у зоні до кута радіогоризонту (~20–40 км від берега), забезпечуючи безперервне відстеження суден у зоні покриття

незалежно від їхніх систем зв'язку. Наприклад, в багатьох портах діють системи VTS (Vessel Traffic Service), що комбінують радарне і візуальне спостереження кораблів. Морські патрулі (фрегати, катери), а також авіаційні засоби дозволяють оперативно перевіряти підозрілі судна.

Проте масштабне патрулювання надзвичайно затратне: одна година польоту патрульного літака C-130 коштує приблизно \$7 500, що в комплексі призводить до мільйонних витрат. Наприклад, оцінка витрат на контроль довгострокової закритої зони показала, що без використання супутників і VMS загальні витрати досягали ~US\$5.6 млн на рік (3000 годин морських патрулів + 350 годин авіації) проти ~US\$0.37 млн при використанні супутників і VMS. Нижче приведені переваги і недоліки:

Переваги: Радар з високою вірогідністю виявляє будь-яке судно (навіть без увімкненого AIS), даючи точний трек його руху. Патрульні одиниці можуть негайно реагувати на загрозу, оскільки об'єкт «на виду». Радари не мають «темних зон» AIS - вони самостійно виявляють об'єкт, незалежно від наявності передавача.

Недоліки: Обмежене зональне покриття. Прибережні радары бачать лише смугу вздовж берега (зазвичай ≤ 40 км), а далекі океанські зони потребують окремих станцій. Морські та повітряні патрулі мають дуже високі поточні витрати (пальне, персонал, обслуговування). Через це постійно патрулювати відкритий океан без залучення інших засобів неможливо.

Стаціонарні сенсори та буйові мережі.

Моніторинг часто здійснюється за допомогою мережі стаціонарних сенсорів. На прибережних станціях або спеціальних буях можуть бути встановлені датчики температури, солоності, розчинного кисню, рівня хлорофілу та нафтопродуктів. Такі системи цілодобово вимірюють стан води і можуть сигналізувати про аварійні зміни параметрів (наприклад, детектори нафтопродуктів миттєво реагують на появу масляної плівки). У деяких країнах застосовують широкі мережі дрейфуючих буйів (ARGO, мережа NOAA NDBC), оснащених гідрологічними сенсорами для

довгострокового спостереження за океанськими течіями та хімічним станом вод [5]. Ці дані допомагають уточнювати поширення забруднень у товщі води, хоча сам по собі рух суден сенсори не відстежують. Нижче приведені переваги і недоліки:

Переваги: Безперервне (реального часу) вимірювання локальних параметрів. Сенсори можуть оперативно фіксувати аварійні концентрації забруднювачів або зміну екологічних показників. Такі мережі ефективні для наукових спостережень і попереджувальних систем.

Недоліки: Дуже обмежене просторове покриття - кожен сенсор охоплює невелику ділянку, тому для масштабного моніторингу потрібна велика кількість станцій. Висока вартість розгортання та обслуговування мережі (забезпечення енергопостачання, захист від корозії та стихії, технічне супроводження). Більшість сенсорів реагують лише на появу забруднення (немає прямого виявлення суден), тому їх використовують як доповнення до активних методів пошуку порушників.

Таким чином, традиційні методи моніторингу доповнюють один одного, але жоден із них сам по собі не є універсальним. У сучасних системах контролю морських вод зазвичай застосовують комбінований підхід: наприклад, поєднують дані з радарів, супутникову інформацію та AIS. Подібне поєднання підвищує ймовірність виявлення порушень, адже інформація з різних джерел доповнює одна одну і зменшує ризик пропуску інцидентів.

1.2 Сучасні технології, роль штучного інтелекту та методи їх застосування

Сучасні технології змінюють підходи до моніторингу акваторій. Глибоке навчання і машинного зору дозволяють аналізувати великі об'єми зображень та відео. Наприклад, система SUBSIM використовує нейромережі для ідентифікації видів за підводними знімками, автоматично створюючи списки фауни, карти покриття водоростей тощо. Розроблено мережі автономних сенсорних вузлів, що застосовують ШІ для обробки даних на місці. Наприклад, система Seabed Sentry (Anduril) складається з мобільних автономних модулів, які самостійно виявляють та передають

критичну інформацію з океанського дна [6]. Ці модулі мають «штучний інтелект» для автономного аналізу та оптимізації зв'язку:

Спостереження з космосу з AI: Використання глибокого навчання при обробці супутникових даних дає змогу швидко виявляти аномалії. Наприклад, європейський проєкт iImagine тренує неймережі на супутникових знімках (Sentinel) для автоматичного виявлення нафтових розливів.

Аналітика AIS: ШІ також застосовують для прогнозування та аналізу поведінки суден. Пропонується поєднувати дані AIS з даними з супутників та планами руху для «безперервного» моніторингу флоту. Як відзначають експерти, «AI-enabled persistent tracking» може компенсувати «сірі» ділянки руху, коли судна відключають AIS.

SUBSIM (Швеція) знаменує собою значний крок вперед у використанні машинного навчання для підводного аналізу зображень, що трансформує дослідження та управління морським середовищем. Завдяки використанню технологій обробки зображень високої чіткості, SUBSIM забезпечує нечувану швидкість і точність обробки даних морського дна порівняно з ручними підходами. Це, наприклад, дуже допомогло у картографуванні морського сміття, такого як мідні бляшанки в Балтійському морі. Такий технологічний прорив дозволяє дослідникам та екологам більш ефективно організовувати свій час та ресурси, а також покращити розуміння складності підводних екосистем. Такі можливості можна розглядати як трансформаційний крок в екологічних дослідженнях не лише на рівні методологічного вдосконалення, але й з огляду на практики управління морськими екосистемами.

У цьому ж напрямку Адміністрація морських портів України (АМПУ) уклала інноваційне партнерство з Національним космічним агентством України з метою впровадження систем супутникового моніторингу в портових зонах [8]. Це партнерство використовує супутникові знімки з високою роздільною здатністю для кращого відстеження та контролю забруднення в межах та навколо портових акваторій. Використовуючи передовий аналіз зображень, система дозволяє точно

визначити джерела забруднення і нанести на карту постраждали райони. Ця здатність забезпечує вибірковий екологічний нагляд, що призводить до більш ефективного і швидкого реагування на випадки забруднення. Такі ініціативи представляють більш широке занепокоєння щодо забезпечення сталого розвитку портів екосистем України, для яких існує потреба у впровадженні інноваційних технологій для вирішення екологічних проблем у морському середовищі.

Передова роль, яку відіграють передові технології в моніторингу навколишнього середовища, зокрема, у випадку виявлення розливів нафти, знайшла своє подальше відображення в проекті iImagine, що фінансується Європейським Союзом. Сервіс WITOil в рамках проекту використовує супутникові знімки та новітні глибокі нейронні мережі для швидкого виявлення нафтових розливів у морських секторах [7]. Окрім виявлення, WITOil надає можливості прогнозування, передбачаючи розвиток і нездатність нафтових розливів до спрямованого руху. Завдяки цим прогнозам команди реагування можуть реагувати з більшою терміновістю і точністю, тим самим зменшуючи шкідливий вплив на навколишнє середовище і рятуючи морські екосистеми. Завдяки поєднанню штучного інтелекту (ШІ) та супутникових технологій WITOil встановлює новий стандарт проактивного моніторингу навколишнього середовища і доводить, що технології, керовані ШІ, можуть вирішувати нагальні проблеми в морському середовищі.

Тонкощі сучасного морського моніторингу тільки посилюються в зонах бойових дій і геополітичного тиску, де відстеження тіньових флотів стикається з особливими труднощами. Судна в таких зонах, як правило, вимикають сигнали автоматичної ідентифікаційної системи (АІС), щоб запобігти виявленню, що ускладнює відстеження морської активності. Щоб вирішити цю проблему, створюються висококласні технології відстеження, які називаються «постійним відстеженням». Ці системи інтегрують дані АІС із захищеними супутниковими мережами, такими як Inmarsat та Iridium, і використовують передову аналітику на основі штучного інтелекту для забезпечення цілодобового моніторингу. Такі

технології, поєднуючи різні джерела даних, дозволяють відстежувати судна, незважаючи на те, що сигнали АІС вибірково вимикаються. Ця здатність не лише ефективно захищає морські судна, але й має вирішальне значення для боротьби з незаконними операціями, наприклад, контрабандою та несанкціонованим виловом риби, що сприяє забезпеченню стабільності в морських зонах.

Ще одна чудова інновація в галузі морського спостереження - це Anduril Seabed Sentry, що являє собою мережу автономних підводних модулів, призначених для моніторингу підводного простору в режимі реального часу. Ці модулі оснащені потужним штучним інтелектом, який дозволяє обробляти дані на місці, не використовуючи стаціонарну інфраструктуру, а отже, безперешкодно обмінюватися даними всередині мережі. Така гнучкість дає можливість системі бути достатньо гнучкою, щоб пристосуватися до різних підводних умов і мати можливість розширювати своє покриття без особливих обмежень. Seabed Sentry особливо практична для захисту важливих об'єктів на морі, в тому числі підводних трубопроводів, кабелів зв'язку і військово-морських об'єктів. Крім того, його здатність відстежувати ситуацію на великих океанських просторах ще більше підкреслює його функціональні можливості як пристрою для нагляду і захисту підводного середовища.

Використання штучного інтелекту для включення його в сучасні системи морського моніторингу має стати зміною парадигми збору, аналізу та застосування даних. Традиційні підходи до моніторингу, хоч і корисні, але недостатньо масштабовані і недостатньо точні, щоб впоратися зі складністю сучасних морських викликів. На відміну від них, системи на основі штучного інтелекту забезпечують автоматизацію, високу точність і обробку великих обсягів даних на місці. Ці можливості роблять їх безцінними в широкому спектрі застосувань - від моніторингу змін навколишнього середовища та збереження морського біорізноманіття до підвищення безпеки та виявлення можливих загроз або незаконних дій. Крім того, адаптивна природа ШІ гарантує, що системи ШІ з часом стають тільки кращими, оскільки вони поглинають нові дані і знання для вдосконалення своєї роботи. Ця

еволюційна здатність робить ШІ основою майбутнього морського спостереження, забезпечуючи ефективні та орієнтовані на майбутнє рішення.

Використання ШІ в моніторингу морських операцій особливо актуальне для захисту навколишнього середовища. Враховуючи незалежність морських екосистем від людської діяльності, важливість наявності ефективних систем моніторингу стає все більш переконливою, оскільки діяльність людини все більше впливає на ці екосистеми. Технології штучного інтелекту дозволяють відчувати невеликі зміни в умовах навколишнього середовища, наприклад, у температурі води, рівні рН або кількості забруднюючих речовин. Ці знання можуть стати рушійною силою природоохоронних програм, які допоможуть мінімізувати наслідки зміни клімату, надмірного вилову риби та інших форм антропогенного тиску на морські екосистеми. Надаючи повну картину динаміки екосистем, системи моніторингу на основі штучного інтелекту допомагають дослідникам і політикам створювати науково обґрунтовані підходи до збереження морських екосистем.

Окрім застосування в екологічній сфері, технології штучного інтелекту також змінюють аспект безпеки в морських регіонах. Можливість інтегрувати та обробляти дані з різних джерел, таких як супутники, дрони та підводні датчики, дозволяє виявляти незаконну діяльність з неперевершеною точністю. Ця здатність особливо корисна для вирішення таких проблем, як піратство, контрабанда та незаконний вилов риби, які серйозно загрожують морській безпеці та економіці. Завдяки підвищенню ситуаційної обізнаності та здатності швидко реагувати на загрози, що розвиваються, системи, керовані штучним інтелектом, підтримують загальну мету створення безпечного та захищеного середовища у водах.

Незважаючи на те, що ШІ має багато переваг для моніторингу вітрильного спорту, існують також важливі етичні та практичні питання, які потребують вирішення. Той факт, що необхідно збирати та аналізувати великі обсяги даних, викликає занепокоєння щодо конфіденційності та безпеки даних, особливо в складних геополітичних умовах. Ефективне використання технологій штучного інтелекту має

ґрунтуватися на сильних регуляторних режимах і прозорих системах управління. Крім того, впровадження систем, керованих ШІ, вимагає значних фінансових і технічних ресурсів, що може бути складним завданням для країн і організацій з обмеженими ресурсами. Ці виклики будуть дуже важливими для того, щоб забезпечити реалізацію максимальних переваг штучного інтелекту з мінімальною ймовірністю потенційних небезпек.

Нарешті, впровадження сучасних технологій, особливо штучного інтелекту, в системах морського моніторингу змінює підходи до вивчення, контролю та захисту морських басейнів. Такі рішення, як SUBSIM, WITOIL та «Вартовий Андурильського морського дна», є яскравою ілюстрацією того, що рішення на основі штучного інтелекту можуть зробити для вирішення складних питань збереження довкілля, безпеки та управління ресурсами. Завдяки автоматизації, точності та масштабованості ці технології відкривають нові перспективи для вивчення та захисту світового океану. Але максимальне використання їхнього потенціалу вимагатиме врахування етичних і регуляторних викликів, логістичних проблем і гарантування відповідального та справедливого використання цих інструментів. З розвитком морського моніторингу функції штучного інтелекту, безумовно, стануть ключовими у визначенні майбутнього взаємодії з морськими просторами та їх захисту.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМАЛІЗАЦІЯ ЗАДАЧІ

2.1 Вибір методів візуального розпізнавання об'єктів на морі

Сьогодні, на тлі активного розвитку морської логістики, захисту державних кордонів та зростання значення екологічної безпеки, особливої актуальності набувають системи морського моніторингу. Безперервний моніторинг руху суден, виявлення підозрілих об'єктів і миттєва реакція на відхилення від норми вимагає використання інтелектуальних інформаційних систем, здатних миттєво обробляти великі обсяги відеоданих [9].

Ця частина документу присвячена розгляду існуючих підходів до морського моніторингу та формулюванню вимог до такої системи, а також чіткому визначенню мети даної роботи. Особливу увагу приділено огляду сучасних методів візуального розпізнавання, які можуть бути застосовані для автоматичного розпізнавання морських об'єктів з використанням комп'ютерного зору та штучного інтелекту.

Для виявлення та класифікації об'єктів на відео й зображеннях використовують сучасні методи комп'ютерного зору та глибокого навчання. Найбільш розповсюдженими моделями нейромережевого розпізнавання є:

YOLO (You Only Look Once) [10].

YOLO (You Only Look Once) - популярна модель виявлення об'єктів, розроблена спеціально для обробки відео в реальному часі. Її основна перевага полягає в здатності виконувати виявлення об'єктів за один прохід, що робить її дуже придатною для завдань, де швидкість є критично важливою. На відміну від інших моделей виявлення об'єктів, які вимагають декількох проходів для аналізу зображення, YOLO розбиває вхідне зображення на сітку і прогнозує обмежувальні рамки та ймовірності класів для об'єктів у кожній комірці сітки, і все це за один прохід. Такий оптимізований підхід дозволяє YOLO досягати чудової швидкості обробки, що робить його найкращим вибором для таких сценаріїв, як аналіз відео в реальному часі та інших додатків у реальному часі [11]. Однією з визначних особливостей YOLO є його універсальність

і простота використання. Завдяки своїй відносно простій структурі YOLO доступна для дослідників і розробників, які працюють у різних галузях. Процес навчання є простим, що дозволяє користувачам адаптувати модель до конкретних випадків використання з мінімальними зусиллями. Від автономних транспортних засобів і систем спостереження до медичної візуалізації та морського моніторингу, YOLO застосовується для вирішення широкого спектру завдань виявлення. Здатність балансувати між швидкістю і точністю зробила його фаворитом серед розробників, які шукають ефективні рішення для виявлення об'єктів.

Особливою сферою, де YOLO досягла успіху, є виявлення морських об'єктів. Вдосконалені версії моделі, такі як YOLOv9 і YOLOv11, продемонстрували виняткову ефективність у виявленні кораблів та інших водних об'єктів на відеозаписах. Ці реалізації були оптимізовані для роботи в динамічному середовищі, що робить їх особливо ефективними для моніторингу в режимі реального часу в таких сценаріях, як охорона портів, відстеження суден і моніторинг навколишнього середовища. Висока швидкість обробки YOLO забезпечує безперервне та надійне виявлення в ситуаціях, де важливим є швидке прийняття рішень. Незважаючи на свої численні переваги, YOLO не позбавлена обмежень. Одна з проблем моделі полягає у виявленні дуже малих або віддалених об'єктів. Оскільки YOLO присвоює прогнози коміркам сітки, менші об'єкти не завжди добре узгоджуються зі структурою сітки, що ускладнює їх точне виявлення. Це обмеження може бути особливо проблематичним у додатках, де розпізнавання дрібних або ледь помітних деталей має вирішальне значення - наприклад, ідентифікація віддалених суден під час морського спостереження або виявлення дрібних тварин під час моніторингу дикої природи.

Ще однією проблемою, яка може вплинути на ефективність YOLO, є перешкоди навколишнього середовища. Наприклад, у морських сценаріях такі фактори, як хвилі, відблиски від сонячного світла на воді або віддзеркалення, можуть вносити шум у дані. Ці умови можуть призвести до помилкових спрацьовувань, коли модель помилково ідентифікує неіснуючі об'єкти, або до помилкових спрацьовувань, коли

реальні об'єкти залишаються невиявленими. Незважаючи на зусилля, спрямовані на підвищення надійності YOLO в таких складних умовах, ці фактори залишаються предметом постійного розгляду для розробників, які використовують модель у складних умовах.

Щоб усунути деякі з цих обмежень, дослідники та розробники продовжують вдосконалювати архітектуру YOLO. Новіші версії моделі включають такі вдосконалення, як покращене виділення об'єктів, вдосконалені стратегії розміщення сітки та передові методи доповнення даних. Ці вдосконалення спрямовані на покращення здатності YOLO виявляти малі або частково затінені об'єкти та зменшити вплив факторів навколишнього середовища на точність виявлення. Крім того, інтеграція YOLO з іншими передовими технологіями, такими як злиття сенсорів і методи попередньої обробки, може допомогти пом'якшити деякі з його слабких сторін, ще більше розширивши його можливості.

Незважаючи на ці виклики, YOLO залишається одним з найбільш надійних і ефективних інструментів для виявлення об'єктів у реальному часі, доступних сьогодні. Баланс швидкості, простоти і продуктивності закріпив за ним місце провідного вибору для застосувань, що вимагають швидкого і надійного виявлення об'єктів. Незалежно від того, чи це моніторинг морського руху, аналіз відеопотоків у реальному часі або забезпечення автономних систем, YOLO продовжує демонструвати свою цінність у сценаріях, де швидкість і точність є критично важливими.

Отже, унікальний підхід YOLO до виявлення об'єктів, що характеризується однопрохідною архітектурою, дозволив їй змінити правила гри в галузі комп'ютерного зору. Хоча система стикається з проблемами при роботі з дрібними об'єктами та шумами навколишнього середовища, постійні вдосконалення та оптимізації продовжують підвищувати її продуктивність. Для розробників і дослідників, які шукають потужне та ефективне рішення для виявлення об'єктів у реальному часі, YOLO залишається головним претендентом, особливо в таких сферах,

як морський моніторинг та інші критичні до швидкості додатки. Його можливості свідчать про потенціал сучасних технологій штучного інтелекту для ефективного і точного вирішення реальних проблем.

Detectron2

Detectron2, розроблена компанією Meta, - це сучасна бібліотека, призначена для вирішення складних завдань виявлення та сегментації об'єктів з винятковою точністю [12]. Побудована на основі таких революційних алгоритмів, як Faster R-CNN та Mask R-CNN, Detectron2 стала незамінним інструментом для дослідників та розробників, які вирішують складні завдання візуального розпізнавання. Його здатність виявляти навіть маленькі, частково затушовані об'єкти вирізняє його з-поміж інших, що робить його особливо корисним для застосувань, де точність має першорядне значення.

Одним з найбільш вражаючих аспектів Detectron2 є його двопрохідна система виявлення та сегментації. На відміну від однопрохідних моделей, таких як YOLO, для яких пріоритетом є швидкість, Detectron2 використовує більш складний підхід, обробляючи зображення в два етапи. Цей метод дозволяє досягти вищої точності ідентифікації об'єктів навіть у зашумованих або складних умовах. Наприклад, у сценаріях, коли об'єкти перекривають один одного або частково приховані, Detectron2 досягає успіху завдяки використанню складних мереж пропозицій регіонів і вдосконалених методів сегментації.

Завдяки своїй точності, Detectron2 особливо добре підходить для галузей, які вимагають ретельного виявлення об'єктів. Наприклад, у медичній візуалізації виявлення тонких аномалій, таких як невеликі пухлини або мікропереломи, вимагає такого рівня точності, який можуть забезпечити лише деякі інструменти, такі як Detectron2. Аналогічно, в промислових інспекціях, де виявлення найдрібніших дефектів у виробничих процесах є критично важливим, ця бібліотека пропонує надійність, необхідну для забезпечення високоякісних результатів. Зосереджуючись на наданні точних результатів, Detectron2 дає можливість фахівцям приймати обґрунтовані рішення на основі високодеталізованого візуального аналізу.

Однак, ця чудова точність не обходиться без компромісів. Detectron2 є інтенсивним в обчисленнях, що вимагає потужного апаратного забезпечення для ефективної роботи [13]. Високопродуктивні графічні процесори часто необхідні для навчання та розгортання моделей, побудованих за допомогою цієї бібліотеки, що може стати бар'єром для невеликих організацій або проектів з обмеженими ресурсами. Крім того, двопрхідний підхід призводить до зниження швидкості обробки порівняно з швидшими однопрхідними альтернативами, такими як YOLO або SSD (Single Shot MultiBox Detector). Для додатків в режимі реального часу, де швидкість так само важлива, як і точність - наприклад, для аналізу відео в реальному часі, автономних транспортних засобів або навігації безпілотників - повільніша продуктивність Detectron2 може не відповідати необхідним вимогам.

Незважаючи на ці виклики, Detectron2 залишається цінним інструментом для широкого спектру застосувань. Його універсальність і модульна конструкція роблять його дуже налаштовуваним, дозволяючи розробникам точно налаштувати моделі для конкретних завдань. Крім того, його відкритий вихідний код гарантує, що він постійно розвивається, а внески світової спільноти дослідників і розробників розширюють його можливості.

Для організацій, які надають перевагу точності над швидкістю, Detectron2 є чудовим вибором. Хоча він не може бути ідеальним рішенням для кожного сценарію, особливо для тих, що вимагають роботи в режимі реального часу, його сильні сторони в наданні точних і надійних результатів неможливо переоцінити. Незалежно від того, чи використовується він для підвищення точності медичних діагнозів, оптимізації промислових процесів або просування досліджень в області комп'ютерного зору, Detectron2 продовжує розширювати межі можливого в області виявлення і сегментації об'єктів.

Отже, Detectron2 - це потужний інструмент, який забезпечує баланс між інноваціями та практичністю, чудово справляючись із завданнями, де точність є ключовим фактором. Хоча його обчислювальні вимоги і низька швидкість обробки

можуть обмежити його застосування в певних середовищах реального часу, його здатність забезпечувати виняткову точність робить його незамінним для багатьох додатків з високими ставками. Для тих, хто прагне використовувати передові технології візуального розпізнавання, Detectron2 - це рішення, яке обіцяє надійність і продуктивність у критично важливих для точності завданнях.

Segment Anything Model (SAM)

Segment Anything Model (SAM) від Meta - це революційний інструмент у галузі сегментації зображень, призначений для точної ідентифікації та ізоляції об'єктів на зображеннях [15]. Відома своєю винятковою продуктивністю та гнучкістю, SAM усуває необхідність додаткового навчання на конкретних наборах даних, що робить її дуже адаптивною для різних випадків використання. Ця найсучасніша модель є значним досягненням у галузі комп'ютерного зору, пропонуючи автоматизовані та інтерактивні режими для задоволення різноманітних потреб користувачів [17].

За своєю суттю, SAM створена для вирішення широкого спектру завдань сегментації, не вимагаючи попередніх знань про конкретні об'єкти, з якими вона зіткнеться. У повністю автоматичному режимі SAM використовує методи попереднього виявлення, такі як обрамлення прямокутниками, щоб окреслити об'єкти на зображенні. Такий підхід дозволяє моделі ідентифікувати і сегментувати об'єкти без втручання користувача. Для більшої точності SAM також підтримує інтерактивне введення даних, де користувачі можуть керувати моделлю, надаючи клацання або підказки, які вдосконалюють процес сегментації. Ця подвійна функціональність робить SAM універсальним інструментом, здатним надавати точні результати як в автоматизованих робочих процесах, так і в сценаріях, керованих користувачем.

Що відрізняє SAM від традиційних моделей сегментації, так це його здатність узагальнювати набори даних. Більшість моделей сегментації вимагають тривалого навчання на конкретних наборах даних, що обмежує їхню застосовність до нових або незнайомих даних. SAM, з іншого боку, попередньо навчений на великому і різноманітному наборі даних, що дозволяє йому добре працювати в різних контекстах.

Така адаптивність робить його цінним ресурсом для галузей і застосувань, де об'єкти і середовище можуть сильно відрізнятися. Незалежно від того, чи це ідентифікація повсякденних об'єктів у споживчих додатках, чи ізоляція складних структур у спеціалізованих галузях, гнучкість SAM гарантує, що вона впорається з будь-яким завданням.

Наприклад, у морській галузі потенціал SAM особливо помітний. Ця галузь часто покладається на аналіз зображень для виконання таких завдань, як моніторинг руху суден, забезпечення безпеки на морі та управління впливом на навколишнє середовище. Традиційні методи виявлення об'єктів можуть ідентифікувати кораблі або судна на зображенні, але їм важко точно визначити їхні контури і межі. Саме в цьому і полягає перевага ЗРК. Покращуючи сегментацію кораблів або інших морських об'єктів, SAM підвищує точність аналізу, що дозволяє покращити моніторинг і прийняття рішень.

Після виявлення судна на знімку з супутника або безпілотної ЗРК може бути використаний для визначення його точної форми і розмірів з надзвичайною точністю. Такий рівень деталізації важливий для таких завдань, як розрахунок розмірів судна, оцінка близькості до інших об'єктів або аналіз моделей руху. Підвищуючи точність сегментації, SAM підтримує більш досконалий аналіз зображень, що, в свою чергу, сприяє підвищенню операційної ефективності та безпеки в морській галузі. Крім того, SAM може допомогти в моніторингу навколишнього середовища, точно сегментуючи такі об'єкти, як розливи нафти або уламки, що дозволяє швидко і цілеспрямовано реагувати на них.

Універсальність SAM робить її цінним інструментом не лише в морській галузі, а й у багатьох інших сферах. В охороні здоров'я вона може допомогти з медичною візуалізацією, ізолюючи пухлини або інші аномалії з високою точністю. У сільському господарстві SAM може допомогти сегментувати посіви або поголів'я худоби на основі аерофотознімків, що допоможе в управлінні ресурсами і моніторингу. У міському плануванні модель може бути використана для аналізу супутникових

знімків міст, виявлення структур і відкритих просторів для інформування проєктів розвитку. Можливості практично безмежні, враховуючи здатність SAM адаптуватися до різних контекстів без необхідності тривалого перенавчання.

Отже, модель Segment Anything Model (SAM) від Meta - це значний крок вперед у технології сегментації зображень. Забезпечуючи виняткову точність і неперевершену гнучкість, SAM спрощує робочі процеси сегментації, одночасно підвищуючи точність у широкому спектрі застосувань. Його адаптивність до різних галузей, від морської до медичної, підкреслює його потенціал як трансформаційного інструменту в комп'ютерному зорі. Поєднання інновацій та надійності SAM, незалежно від того, чи використовується він для автоматизованих завдань, чи вдосконалюється за допомогою користувача, змінює правила гри для всіх, хто працює з даними зображень.

Інші сучасні архітектури

Трансформаторні детектори, такі як DETR і Swin Transformer, а також такі моделі, як EfficientDet, представляють собою значний прогрес у комп'ютерному зорі завдяки поєднанню унікальних переваг згорткових нейронних мереж (CNN) і трансформаторів [18]. ШНМ добре відомі своєю здатністю захоплювати дрібні просторові деталі, що робить їх ідеальними для ідентифікації текстур, країв та інших складних об'єктів на зображеннях. З іншого боку, трансформери чудово моделюють довгострокові залежності та захоплюють глобальний контекст, що є важливим для розуміння взаємозв'язків між об'єктами на зображенні та аналізу сцени в цілому. Інтегруючи ці сильні сторони, ці моделі досягають чудової продуктивності в таких завданнях, як виявлення об'єктів і сегментація зображень, встановлюючи нові стандарти точності та універсальності.

DETR (DEtection TRansformer) використовує механізм самоуваги, властивий трансформаторам, для безпосереднього прогнозування місцезнаходження і класифікації об'єктів, усуваючи необхідність в традиційних компонентах, таких як якірні коробки, які зазвичай використовувалися в попередніх моделях виявлення.

Swin Transformer, ієрархічна модель трансформатора, підвищує ефективність завдяки обробці зображень у менших вікнах, що не перекриваються, зберігаючи при цьому можливість захоплення глобального контексту. Аналогічно, EfficientDet представляє масштабовану архітектуру, яка балансує між точністю і обчислювальною ефективністю, використовуючи складне масштабування для оптимізації вхідної роздільної здатності, глибини і ширини по всій мережі.

Ці інновації значно покращили можливості моделей комп'ютерного зору, дозволивши їм добре працювати з різноманітними наборами даних і складними сценаріями. Однак цей стрибок у продуктивності має свою ціну. Моделі на основі трансформаторів є обчислювально інтенсивними і потребують висококласного обладнання, такого як графічні процесори зі значними обсягами пам'яті, для ефективного навчання та розгортання. Їх залежність від передових архітектур також призводить до складності в налаштуванні, часто вимагаючи стрімкої кривої навчання для користувачів без сильної технічної підготовки.

Для дослідників і організацій, які мають доступ до необхідних обчислювальних ресурсів, ці моделі відкривають захоплюючі можливості для розвитку додатків у різних сферах - від автономних транспортних засобів і медичної візуалізації до відеоаналізу і доповненої реальності. Однак для користувачів з обмеженим апаратним забезпеченням або досвідом ці моделі можуть створювати значні бар'єри для входу на ринок. Вирішення цих проблем вимагатиме постійних зусиль з оптимізації обчислювальної ефективності та спрощення процесів впровадження, що зробить такі передові технології доступнішими для ширшої аудиторії. Незважаючи на ці перешкоди, детектори на основі трансформаторів залишаються революційною інновацією, яка прокладає шлях до більш надійних і універсальних систем комп'ютерного зору.

Кожен метод розпізнавання об'єктів у морському середовищі має свій набір переваг і обмежень, особливо при вирішенні унікальних завдань, пов'язаних з цим специфічним контекстом. Морське середовище створює складні і динамічні умови для

систем комп'ютерного зору, що характеризуються монохроматичним, постійно рухомим фоном, постійною хвильовою активністю, відблесками від сонця і водної поверхні, а також частою появою невеликих, віддалених об'єктів. Ці фактори разом утворюють дуже складну робочу область, де досягнення балансу між швидкістю обробки і точністю виявлення має вирішальне значення для успіху будь-якої системи моніторингу або виявлення.

Враховуючи ці обмеження, для дослідження було обрано YOLOv11 (You Only Look Once, версія 11) - однопрохідну модель глибокого навчання, відому своєю здатністю виконувати виявлення об'єктів у реальному часі з мінімальною затримкою. YOLOv11 вирізняється своєю оптимізованою архітектурою, розробленою для обробки візуальних вхідних даних за один обчислювальний прохід, що забезпечує швидке виявлення без значного зниження точності. Така ефективність робить його особливо придатним для сценаріїв, де швидке прийняття рішень і негайне реагування мають першорядне значення. Хоча альтернативні підходи, такі як двопрохідні детектори або сегментатори, такі як Faster R-CNN або SAM (Segment Anything Model), пропонують більшу точність у певних ситуаціях, їхня обчислювальна складність призводить до значного зниження продуктивності системи. Ці повільніші методи особливо погано підходять для застосування в режимі реального часу, де затримки в обробці даних можуть підірвати ефективність системи моніторингу.

У контексті систем морського моніторингу, де швидке виявлення і реагування є важливими, YOLOv11 являє собою оптимальний компроміс між обчислювальною продуктивністю, точністю виявлення і простотою інтеграції в існуючі структури. Його універсальність і адаптивність дозволяють йому відповідати вимогам виявлення об'єктів у реальному часі навіть у несприятливих умовах навколишнього середовища. Хоча основна роль YOLOv11 полягає в тому, щоб слугувати базовим інструментом виявлення, за потреби його можна доповнити додатковими алгоритмами для уточнення контурів об'єктів або покращення аналізу після виявлення. Такий модульний підхід забезпечує гнучкість у вирішенні конкретних завдань або

покращенні результатів без зайвого обчислювального навантаження на початковому етапі виявлення.

Зрештою, здатність YOLOv11 збалансувати швидкість і точність обробки в поєднанні з можливостями роботи в режимі реального часу робить його ідеальним вибором для первинного виявлення в дуже мінливих і складних умовах морського середовища. Ця робота ілюструє, як YOLOv11 може бути успішно використаний для задоволення експлуатаційних вимог таких систем, пропонуючи практичне рішення технічних проблем, що виникають у морському середовищі.

2.1 Постановка задачі системи моніторингу

Основною метою системи моніторингу є постійне спостереження за морськими регіонами, захист державних кордонів і припинення незаконних операцій, таких як контрабанда, незаконний вилов риби або несанкціоноване перебування суден у наших водах. Завдяки передовим технологіям система прагне надавати регулярні оновлення тим, хто керує системою, і використовувати автоматизацію, щоб зробити робочі процеси більш ефективними і з меншою кількістю помилок.

У морському середовищі багато простору і багато непередбачуваних проблем. Хороше розуміння і управління таким місцем залежить від систем, які сканують інформацію в режимі реального часу і пропонують рішення. Планується, що проектована система моніторингу буде поєднувати радар, відеоспостереження, супутникові знімки і штучний інтелект, пропонуючи своїм операторам комплексне уявлення про те, що відбувається в морській акваторії.

Детекція об'єктів (виявлення суден)

Основним завданням системи є автоматична ідентифікація кораблів у регіоні спостереження. Всі наступні дії побудовані на виявленні об'єктів, що дозволяє системі знаходити цілі і вивчати, що вони собою являють. Для цього вона використовує передові методи обробки, а також різноманітні вхідні дані з датчиків.

Щоб отримати точну інформацію, система зчитує дані з радарів, камер і супутникових знімків. Використання цих датчиків разом дає детальну картину моря, яка дозволяє виявляти об'єкти, незалежно від розміру, в різних умовах. Завдяки цим складним алгоритмам супутники можуть виявляти судна навіть там, де відсутня система AIS. З цієї причини вона допомагає відстежувати і перехоплювати будь-які судна, які не можуть чітко визначити своє місцезнаходження. Система побудована таким чином, що зростаючі обсяги даних не становлять жодних проблем для моніторингу великої чи малої зони спостереження. Як результат, технологія може використовуватися як у переповнених морських районах, так і у відкритому морі.

Ідентифікація об'єктів

Після виявлення суден не менш важливо знати, що вони собою являють. Система використовує комп'ютерний зір для оцінки зовнішнього вигляду човнів на зображеннях з відео або супутників. Після цього ознаки зіставляються з наявною базою даних суден для сортування виявленого об'єкта. Як приклад, система здатна сортувати судна на такі категорії, як вантажні, пасажирські, рибальські або рекреаційні судна.

При наявності доступу до потрібних ресурсів, система зможе розпізнавати судна, які не передають дані за допомогою AIS або навмисно вимкнули свою AIS. Система використовує фотографії і радар, щоб відстежувати будь-які підводні човни без екіпажу, що унеможливорює їхнє непомічання.

Програмне забезпечення для ідентифікації пов'язане з відомими морськими базами даних, що підвищує надійність його результатів. Цей процес дозволяє операторам ідентифікувати відомі об'єкти за наявними у них даними, підвищуючи обізнаність про те, що відбувається.

Трекінг суден.

Безперервне відстеження суден - останній елемент системи моніторингу. Виконуючи це завдання, ми миттєво відстежуємо переміщення кожного судна, тому оператори завжди можуть бачити, що відбувається у воді. Безпечне судноплавство,

виявлення ризиків на воді та прогнозування руху суден - все це стає можливим завдяки відстеженню суден.

Завдяки цій функції команди безпеки можуть помітити потенційні зіткнення, коли транспортні засоби відхиляються від запланованих маршрутів або коли транспортні засоби блукають у підозрілих місцях.

Етичні міркування

Хоча нова система моніторингу забезпечить багато переваг, слід приділити ретельну увагу її етичному впливу. Викликають занепокоєння питання конфіденційності, безпека даних і можливість зловживання системою. Щоб контролювати ці ризики, система повинна завжди дотримуватися важливих етичних стандартів.

Конфіденційність даних означає, що приватні дані залишаються в безпеці і доступні лише тим, хто повинен їх бачити. Забезпечення нагляду за установами та їхньої відповідальності є пріоритетом, щоб їхні дії не виходили за рамки дозволеного.

Запланована система морського моніторингу полегшить забезпечення безпеки, захист від несанкціонованих дій і охорону морських кордонів. Завдяки автоматизованому виявленню, ідентифікації та моніторингу суден система сприяє підвищенню операційної ефективності і надає операторам миттєву інформацію про те, що відбувається у воді. Поєднання високотехнологічних інструментів та високих етичних норм допоможе системі значно покращити безпеку та управління на морі в майбутньому.

2.3 Вимоги до системи моніторингу та розпізнавання суден

Щоб системи працювали добре, могли розвиватися в міру необхідності та мали високу якість, вони потребують як функціональних, так і нефункціональних вимог під час проектування та експлуатації. Ці настанови пояснюють, чого прагне досягти система і як вона повинна працювати, а також створюють стабільний план для її ефективного розвитку та використання.

Функціональні вимоги детально пояснюють завдання, які система повинна виконувати для досягнення своєї мети. До них відносяться:

Детекція і класифікація суден.

Розпізнавання та ідентифікація суден шляхом виявлення та класифікації є важливим. Будь-яка система повинна реєструвати всі судна, що знаходяться в межах визначеної зони контролю. Окрім ідентифікації суден, вона також повинна розрізняти вантажні судна, пасажирські судна, рибальські човни тощо. Ефективне виконання цих завдань залежить від використання цієї технічної можливості.

Журналювання та звітність.

Система повинна повідомляти про місцезнаходження суден, напрямок їхнього руху і коли спрацьовують сигнали тривоги. Автоматична реєстрація необхідна для того, щоб згодом можна було перевірити і переглянути повний запис операцій. Крім того, інструмент повинен дозволяти користувачам генерувати детальну інформацію, яка сприяє прийняттю рішень на основі даних і відповідає нормативним правилам.

Нефункціональні вимоги зосереджені на якісних характеристиках продукту. Вони підтверджують, що система є ефективною та надійною в усіх умовах, з якими вона може зіткнутися:

Продуктивністю системи та час, необхідний для реакції.

Для швидкого отримання точних кінцевих результатів система повинна обробляти дані одразу після їх збору. Щоб мінімізувати затримки, система повинна миттєво відображати зміни в положенні суден на своєму інтерфейсі. Вона також

повинна обробляти швидкі потоки даних, наприклад, відео, зі швидкістю 25-30 кадрів в секунду, щоб моніторинг залишався безперервним і плавним.

Чи можливо тут досягти точності і надійності? Так, але судно має бути завжди точно виявлене і класифіковане. Щоб зробити систему надійною, вона повинна гарантувати, що буде дуже мало пропущених виявлень і дуже мало помилкових викликів. Крім того, енергетичні системи повинні працювати постійно, вдень і вночі. Щоб зробити систему більш надійною, вона повинна бути оснащена системою аварійного відновлення, щоб вона швидко перезапускалася після проблем або збоїв.

Масштабованість і розширюваність є важливою характеристикою.

Оскільки морський трафік і зони спостереження змінюються, система повинна розширюватися, щоб не відставати. Система з модульною архітектурою є корисною, оскільки дозволяє легко підключати нові датчики, алгоритми або модулі. Завдяки своїй гнучкості система може впоратися з новими обов'язками і технічними оновленнями.

3 РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ МОНІТОРИНГУ ОБ'ЄКТІВ МОРСЬКОЇ АКВАТОРІЇ

3.1 Архітектура системи моніторингу об'єктів морської акваторії

Архітектура розробленої системи відповідає багаторівневому підходу до організації відеоспостереження з аналітикою. Логічно систему можна уявити як сумісну структуру з декількох рівнів: рівень збору даних (камери), рівень передачі (мережа зв'язку), рівень обробки та аналітики (сервер, алгоритми ШІ) і прикладний рівень (інтерфейс користувача, база даних).

За аналогією з класичними системами відеоспостереження, розроблена структура складається з таких компонентів: IP-камери для захоплення відео, мережеві засоби передачі даних (Ethernet/Wi-Fi) для передавання відеопотоку на сервер, серверні модулі обробки (компоненти OpenCV, моделі ШІ) та клієнтські модулі для відображення результатів і аналітики. Таке розмежування етапів відповідає загальновідомій схемі морського спостереження:

- збір даних у складних умовах,
- обчислювальна обробка даних,
- формування прикладного результату після обробки.

На рівні технічної архітектури камера надсилає відеопотік по протоколу RTSP (Real-Time Streaming Protocol), що забезпечує керування потоком (відтворення/паузу) без перенесення медіаданих через RTSP. Зазвичай IP-камери з підтримкою RTSP передають стиснений відео- та аудіопотік (часто в кодеку H.264) на сервер або мережевий відеореєстратор (NVR) для подальшої обробки. Використання RTSP дозволить масштабувати систему (додавати камери) та здійснювати віддалений моніторинг у реальному часі.

У нашій системі мережевий сервер (наприклад, з GPU-акселератором) отримує кадри від камер, виконує попередню обробку та детекцію. Результати класифікації зберігаються в базі даних або проміжному сховищі, а також транслюються до

користувачького інтерфейсу. Клієнтський модуль (веб-додаток на Flask або Streamlit) підтягує ці дані для візуалізації. Такий підхід відповідає типовій архітектурі систем відеоспостереження, що складаються з камер, пристроїв запису, мережі, моніторингу та аналітики.

Основні компоненти системи:

Мережеве з'єднання виконує роль моста для передачі відеопотоку від джерела до сервера. Для забезпечення безперебійної та надійної роботи системи використовується кабельне з'єднання Ethernet, що гарантує швидку та стабільну передачу інформації.

Сервер обробки зберігає всі завдання і збирає дані, необхідні для аналізу. Багаторазові операції обробляються за допомогою потужних і новітніх інструментів і фреймворків. Це досягається двома способами: за допомогою OpenCV для стабілізації руху і зображення та організації схеми, а також для виявлення і категоризації об'єктів за допомогою моделі YOLOv11 з бібліотеки YOLO. Сервер також надає аналітичну інформацію з усіх оброблених відеопотоків. Це важливо для гарантування точної та швидкої роботи системи.

Всі типи інформації, що використовуються в системі, зберігаються з використанням потужного підходу до зберігання даних. Це означає наявність місця, де можна безпечно зберігати зображення, виявлення та журнали. Оскільки система зберігання зберігає важливі дані для пошуку, це робить систему більш надійною та контрольованою.

Графічний інтерфейс надає користувачам простий інтерфейс для підключення до всіх функцій системи. Створений за допомогою Flask як веб-додатку, інтерфейс показує відеопотік з об'єктами, які чітко пронумеровані. Аналітичні звіти також включені, даючи чітке резюме даних тестування, які може використовувати будь-хто. Завдяки інтерфейсу користувачі не мають проблем з використанням системи, оскільки він пов'язує внутрішні дії з зовнішнім інтерфейсом.

Схематично наша система поєднує трирівневу модель: фізичний (камери та зв'язок), обчислювальний (обробка відео та інференс нейромережі) і прикладний (інтерфейс та аналітика). Після попередньої обробки кадрів від камер вони надходять у модель детекції, результати якої акумулюються і виводяться користувачу. Такий дизайн забезпечує модульність: наприклад, заміна камери або зміна моделі ШІ не вимагає перебудови всієї системи.

3.2 Підсистема збору та обробки відеоданих із камер

Підсистема збору відео в реалізованому рішенні реалізована на основі офлайн-відео, отриманих з платформи YouTube. Такий спосіб виникає через обмеження доступу до живих RTSP-потоків у реальному часі. Для отримання відеофайлів використовується інструмент FFmpeg - потужна консольна утиліта для обробки мультимедійних даних. FFmpeg дозволяє завантажувати відео, конвертувати та розбивати його на окремі кадри, які аналізуються системою.

Все починається з отримання посилання на відео з платформи YouTube. Потім FFmpeg використовується для вилучення фрагментів відео або цілого відео у форматі .mp4 з обраною частотою кадрів (наприклад, 25 або 30 FPS) і роздільною здатністю (наприклад, 1280×720 px). Завдяки такому підходу ви зможете працювати з численними сценаріями (день, ніч, зйомка з дрону) без необхідності реального підключення до камер.

На рівні програмної реалізації обробка відео виконується за допомогою OpenCV (cv2). Завантажений відеофайл завантажується в cv2.VideoCapture, який дозволяє передавати зображення для аналізу кадр за кадром. Кожен кадр BGR перетворюється в RGB або відтінки сірого для підготовки вхідної нейронної моделі. Обробка відбувається послідовно: кадри зчитуються, змінюються до певного розміру (наприклад, 1280×720 px) і передаються на детектування.

Таким чином, за відсутності потокового RTSP-відео використання відеофрагментів з YouTube дає нам можливість повноцінно імітувати роботу системи

моніторингу в реалістичних умовах. Такий підхід також полегшує тестування моделі з великою кількістю різних налаштувань - у різний час доби, з різними типами суден і в різних локаціях.

Нормалізація та масштабування

Кадри зменшуються до розміру, необхідного для моделі, що пришвидшує обробку та зменшує навантаження на GPU. Одночасно виконуються перетворення формату кольорів (BGR→RGB) та нормалізація значень пікселів.

Зниження шуму

Використовуються фільтри для розмивання зображення (смукування) на основі ковельних операцій. Згідно з документацією OpenCV, розмивання зображення низькочастотним фільтром корисне для усунення шуму, оскільки воно усуває високочастотний вміст (наприклад, шум) зображення. Для цього застосовуються функції `cv2.GaussianBlur()`, `cv2.medianBlur()` або `cv2.bilateralFilter()`, що згладжують випадкові шуми, зберігаючи при цьому контури об'єктів.

Стабілізація відео

Фіксовані встановлені камери можуть незначно «тремтіти» від вітру, коливань або механічних коливань кріплення. Навіть незначні зміщення призводять до помітних зсувів зображення. Для компенсації цього застосовується алгоритм стабілізації. Типовий підхід - пошук фіксованих орієнтирів (наприклад, контурів берегової лінії або стійких об'єктів) за допомогою оператора `Canny`, після чого обчислюється 2-D перетворення (зсув) між кадрами. Як показано у методі стабілізації для прибережних відеосистем, після виділення країв камерою `Canny` вимірюють двовимірне зміщення поточного кадру відносно опорного за допомогою субпіксельної крос-кореляції. Цей зсув застосовується для вирівнювання кадру. Було застосовано спрощений варіант: виконуємо Gaussian-розмивання для зменшення шуму, виділення ключових особливостей (натисних точок) за допомогою `cv2.goodFeaturesToTrack()`, після чого обчислюємо афінне перетворення на основі найдовірніших пар точок.

Корекція освітленості.

Іноді виконується вирівнювання гістограми або адаптивна гістограма (CLANE) для покращення контрастності кадрів за умов поганого освітлення (туман, сутінки). Це підвищує вірогідність правильного виявлення об'єктів при складних оптичних умовах. Кожен оброблений кадр потім передається до модуля детекції. Система реалізована таким чином, що камера постійно надсилає фрейми, а сервіс обробки опитує їх циклічно або за подіями, забезпечуючи аналіз відео в реальному часі. Після детекції результати (координати та клас об'єкта) асоціюються з кожним кадром і можуть зберігатися для подальшої аналітики (наприклад, трасування траєкторії судна, підрахунок кількості об'єктів за одиницю часу тощо). Попередня обробка виявилася критичною для зниження кількості хибних спрацьовувань детектора: згладжене і вирівняне зображення дозволяє моделі точніше локалізувати реальні об'єкти, м'яко фільтруючи випадкові шуми.

3.3 Алгоритми виявлення та класифікації суден на зображеннях

Для виявлення суден в обробленому відео обрана модель **YOLOv11** від Ultralytics. Це сучасна версія сімейства *You Only Look Once* (YOLO), спроектована для підвищення продуктивності детекції у реальному часі. Як зазначено в документації Ultralytics, YOLOv11 використовує покращену архітектуру backbone і neck, є *безанкоровою* (anchor-free) та забезпечує оптимальний баланс між точністю і швидкістю. Наприклад, найменша версія YOLOv11 демонструє mAP 37.3 на наборі COCO та час обробки 0.99 мс на GPU A100. Вибір YOLOv11 обґрунтовано високою швидкістю інференсу і легкою інтеграцією в Python-середовище: за допомогою бібліотеки Ultralytics виклик моделі для виявлення можна здійснити кількома рядками коду [19].

Моделі навчали на власноруч розміченому наборі зображень морської акваторії з суднами. Розмітка виконувалася у форматі, сумісному з YOLO: для кожного зображення створювався текстовий файл .txt, де кожен рядок містить номер класу (для

нашої системи —, наприклад, **0** = «корабель»), центр і ширину/висоту об'єкта в нормалізованих одиницях (відносно розмірів зображення).

Формально, формат рядка: class_id center_x center_y width height. Наприклад: 0 0.523 0.412 0.122 0.089 означає один об'єкт класу 0 із центром у точці (0.523·ширина, 0.412·висота) і габаритами 0.122 і 0.089 (у частках від розмірів зображення). Конфігураційний YAML-файл даних описує шляхи до директорій з навчальною та валідаційною підвибірками, а також список імен класів, для декількох класів суден

```
model = YOLO("runs/train/custom_model4/weights/best.pt")

1 usage
def process_frame(frame):
    results = model.predict(frame, conf=0.3, verbose=False)[0]

    if results.boxes is not None:
        for box in results.boxes:
            x1, y1, x2, y2 = box.xyxy[0].cpu().numpy()
            cls = int(box.cls[0].cpu().numpy())
            conf = box.conf[0].cpu().numpy()
            label = f'{results.names[cls]} {conf:.2f}'
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
            cv2.putText(frame, label, org=(int(x1), int(y1) - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.6, color=(0, 255, 0), thickness=2)

    return frame
```

Рис 3.3.1 Приклад ініціалізації та використання моделі YOLOv11 на Python

У цьому прикладі зображення frame.jpg піддається інференсу, а results містить об'єкти з інформацією про детекцію (координати, класи, ймовірності).

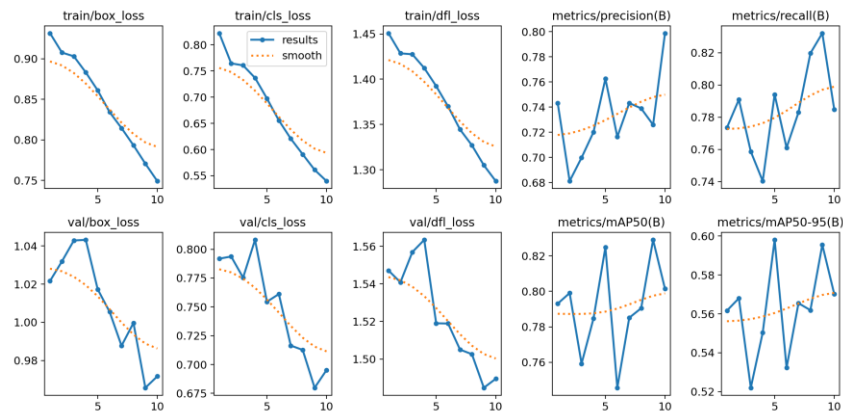


Рис 3.3.2 results.png

Для навчання моделі на нашому власному наборі зображень використовувався метод Ultralytics `model.train()`. Команда навчає модель yolo на даних, що описані у файлі `marine_data.yaml`, протягом 20 епох. Під час тренування Ultralytics виводить метрики точності (mAP) та втрат (loss). У разі потреби можна відрегулювати параметри (learning rate, batch, шари) через аргументи методів або CLI.

```
from ultralytics import YOLO

model = YOLO("yolo11n.pt") # або yolov8m.pt / yolov8s.pt / yolov8x.pt

# Тренування
model.train(
    data="data.yaml",
    epochs=20,
    imgsz=640,
    batch=16,
    project="runs/train",
    name="custom_model",
    resume=False
)
```

Рис 3.3.3 Приклад ініціалізації та навчання моделі YOLOv11 на Python

Згідно з дослідженнями, сучасні моделі на основі згорткових мереж, зокрема YOLOv11, забезпечують високу точність та швидкість при детекції суден. Крім того, Ultralytics активно підтримує пакет, що спрощує інтеграцію та оновлення моделі. Вибір обґрунтовується також тим, що YOLOv11 показує хороші результати в умовах обмежених ресурсів: він досягає прийнятної точності при мінімальній затримці, що критично для реального часу.

Результати роботи алгоритму детекції продемонстровані на рисунку 3.3: на цьому прикладі кадру відео з камери нанесені прямокутники на всіх ідентифікованих судах із зазначенням класу (корабель) і ймовірності (confidence).

Алгоритм коректно виділив плавзасоби у кадрі, незважаючи на хвилі та зміну освітлення.

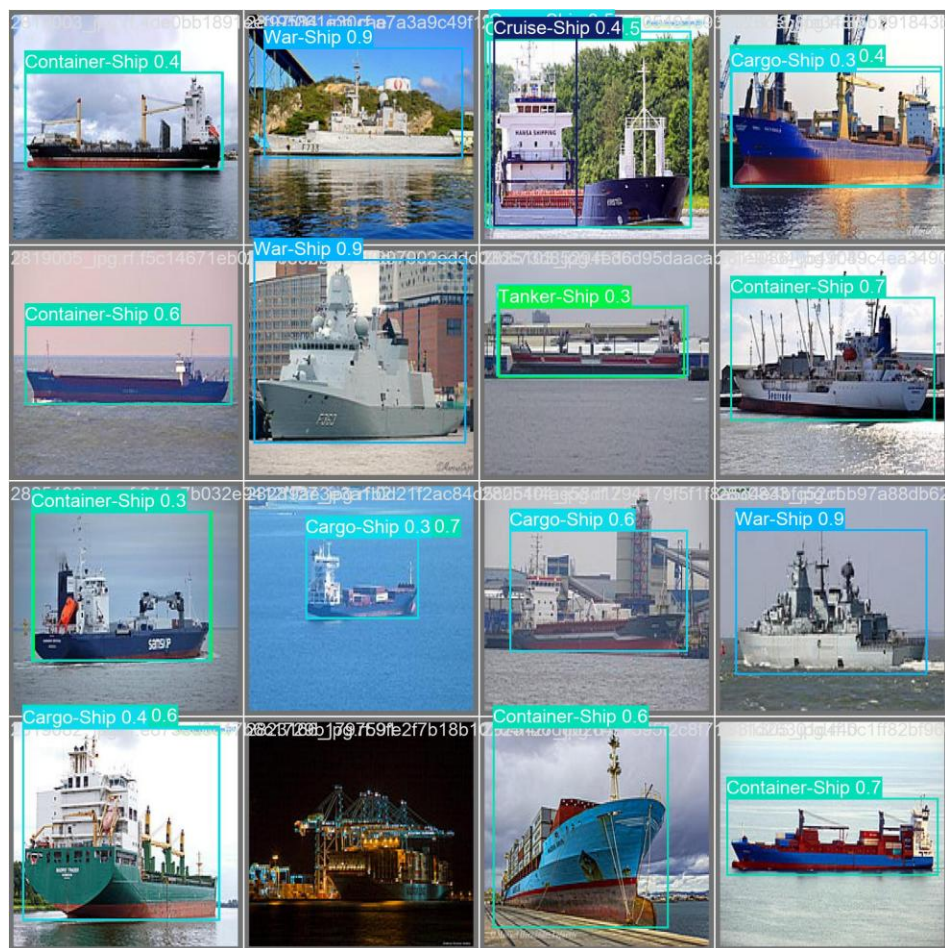


Рис3.3.4 зображення того як алгоритм виділив плавзасоби у кадрі, незважаючи на хвилі та зміну освітлення

3.4 Інтеграція алгоритмів у єдиний програмний комплекс

Для об'єднання всіх компонентів у робочий прототип був розроблений веб-додаток із використанням Flask (можна також використовувати Streamlit як альтернативу). Серверна частина відповідає за отримання кадрів з камер, передачу їх у модель детекції, а клієнтська - за відображення відеопотоку з результатами.

Роутинг і потоки даних

Створюються HTTP-маршрути, наприклад: "/" для основної сторінки та "/video_feed" для трансляції відеопотоку. Для Flask маршрут "/video_feed" повертає Response зі спеціальним MIME-типом multipart/x-mixed-replace, що дозволяє передавати послідовність JPEG-кадрів у режимі реального часу.

```
2 usages
def gen_frames(path):
    cap = cv2.VideoCapture(path)
    while True:
        success, frame = cap.read()
        if not success:
            break
        annotated = process_frame(frame)
        ret, buffer = cv2.imencode(ext='.jpg', annotated)
        frame_bytes = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n')
    cap.release()

@app.route('/video_feed')
def video_feed():
    filename = session.get("last_video", None)
    if filename:
        path = os.path.join("static", filename)
        return Response(gen_frames(path), mimetype='multipart/x-mixed-replace; boundary=frame')
    else:
        return "Відео не знайдено", 404
```

Рис 3.4.1 генераторна функція, яка зчитує кадри з камери, виконує детекцію і повертає серійну відповідь

До кожного кадру застосовується модель `model(frame)`, після чого результат відображається (метод `plot()`) прямо на зображенні. Заштрихований кадр кодується у JPEG і надсилається у складі HTTP-відповіді з потрібними заголовками. Таким чином,

клієнтський браузер постійно отримує оновлений потік із кадрами. Подібний підхід базується на принципі *streaming response* у Flask

Інтерфейс користувача

Головна сторінка веб-додатку містить відеопотік і елементи управління. Якщо використовується Flask, то зручно будувати HTML-шаблони з `index.html`, куди вставляється тег `` для показу відео. У Streamlit можна використовувати компонент `st.image()` або плагіни для показу webcam-стріму.

Streamlit спрощує інтерфейс: він дозволяє легко додати кнопки, вибір відеоджерела, а також інтерактивні графіки. У нашому прототипі ми реалізували інтерфейс таким чином, щоб користувач міг бачити поточний відеопотік з позначеними об'єктами (через прямокутники і наклейки класу на зображенні).

Передача та обробка результатів

Після детекції координати та класи об'єктів можуть передаватися до сховища або на окремий сервіс для аналізу. У простому прототипі ми зберігаємо результати у вбудовану у пам'ять структуру (наприклад, список або базу SQLite). Кожний детектований корабель при отриманні кадру записується з часовою міткою і може бути ідентифікований (наприклад, шляхом прив'язки номера до траєкторії).

На основі цих даних формується аналітика: за певний проміжок часу підраховується загальна кількість кораблів, визначаються найчастіші клас-типи, відображаються теплові карти активності в місцях фіксації руху і т.д. Можливе також налаштування сповіщень (`alert`) - наприклад, якщо певний район акваторії порушено рухом, або виявлено нетипові об'єкти.

Моніторинг продуктивності

У реальному часі важливо контролювати, щоб система витримувала навантаження. Вимірюються показники FPS (кадрів за секунду), затримка обробки кадру (`latency`), завантаження CPU/GPU. Наш прототип показує близько 30 FPS на звичайному CPU середнього класу (AMD Ryzen 7700x3d) при розмірі кадру 1280×780 і одному потоці відео, що достатньо для спостереження більшості прибережних зон.

Інтеграція Flask незначно додає накладні витрати на передачу даних по HTTP, але загалом пропускає потік з мінімальною затримкою. У майбутніх версіях можна застосувати черги та асинхронну обробку (наприклад, через Redis або Celery) для масштабування на кілька камер.

Таким чином, розроблена система складається з взаємодіючих модулів: веб-сервер (Flask/Streamlit) + обробка відео (OpenCV) + детекція (YOLOv11) + сховище/інтерфейс. Комунікація між ними забезпечує передачу потоку кадрів і результатів (наприклад, через HTTP або WebSocket, залежно від обраного фреймворку). Така архітектура є завершеним програмним комплексом, який готовий до подальшого тестування у реальних або змодельованих умовах.

3.5 Експериментальна перевірка та оцінка ефективності системи

Для перевірки ефективності системи моніторингу суден було проведено низку експериментів в умовах, максимально наближених до тих, що зустрічаються під час звичайної експлуатації суден. Випробування системи в реальних ситуаціях дозволило дослідникам перевірити її практичну цінність і надійність. Метою тестування було зрозуміти, що працює добре, а що ні при використанні глибокого навчання для виявлення суден. Експерименти також показали, наскільки програмне забезпечення відповідає функціональним і нефункціональним вимогам, пов'язаним з точністю, продуктивністю і надійністю системи.

Вісім ключових типів вхідних даних були використані в експериментальній установці для перевірки широти та універсальності системи. Зокрема, дані, використані для цього експерименту, включали статичні зображення, збережені відеофайли та відео, завантажені з таких сайтів, як YouTube. Кожен тип вхідних даних оброблявся автоматично, а результати були доступні на корисній веб-сторінці. Завдяки такому інтерфейсу кінцеві користувачі могли без зусиль використовувати і розуміти результати роботи системи.

Точність виявлення, продуктивність і стабільність системи оцінювалися під час різних сценаріїв. Завдяки оцінці були зроблені висновки щодо продуктивності всієї запропонованої архітектури системи та аспекту машинного навчання. Крім того, результати дали вказівки щодо покращення роботи системи та вирішення проблем, виявлених під час тестування.

3.5.1 Опис експериментального середовища та тестових сценаріїв

З метою оцінки працездатності та ефективності запропонованої системи моніторингу суден було проведено експериментальне тестування в умовах, максимально наближених до реального застосування. Для цього було створено веб-додаток з використанням мови програмування Python та фреймворку Flask, який забезпечує взаємодію користувача з моделлю глибокого навчання YOLOv11.

Модель була попередньо навчена на спеціалізованому датасеті суден, що забезпечує розпізнавання об'єктів на зображеннях та у відеопотоці з високою точністю.

- Технічне середовище тестування включало:
- Операційна система: Windows 10 x64
- Процесор: AMD Ryzen 5 5800X3D
- Відеокарта: Sapphire Radeon RX 6650 XT (16GB)
- ОЗП: 64 ГБ DDR4
- Фреймворк: Flask 2.3.3
- Бібліотеки: OpenCV, Ultralytics YOLOv11, yt-dlp, ffmpeg
- Браузер: Google Chrome (для взаємодії з веб-інтерфейсом)

Тестові сценарії були розроблені для перевірки наступного функціоналу:

Завантаження та обробка статичних зображень: перевірка можливості користувача надіслати зображення з судном через форму, відображення результату з накладеними рамками та класами виявлених об'єктів.

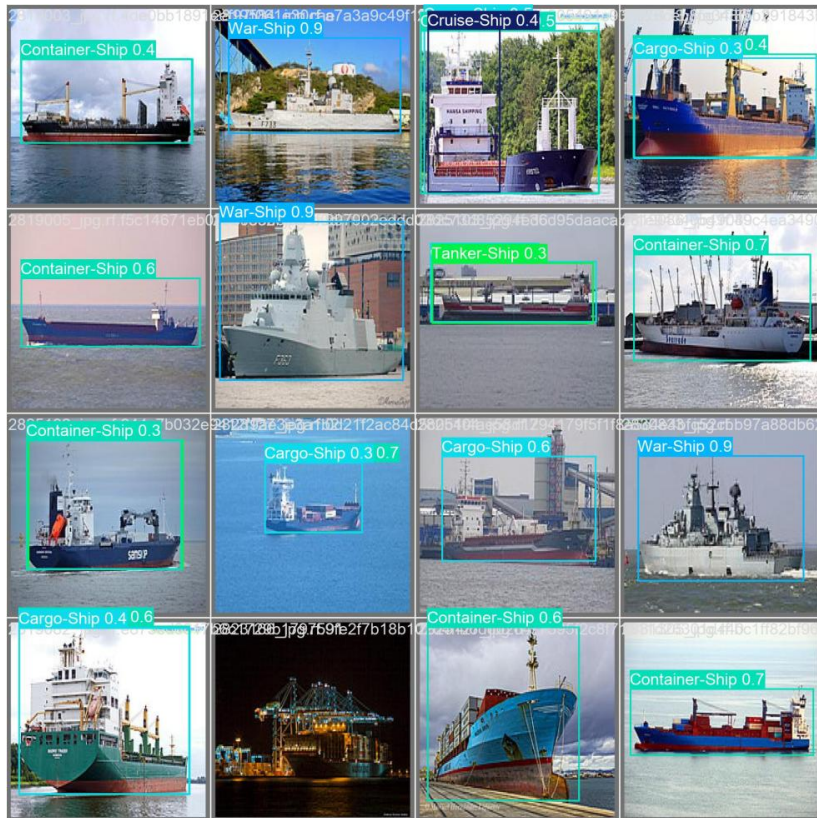


Рис 3.5.1 демонстрація аналізу статичних зображень

Обробка відео з YouTube: система дозволяє вставити посилання на відео, яке автоматично завантажується, зберігається з унікальним іменем, проходить обробку, а результат виводиться у реальному часі через вбудований стрім на тій самій сторінці.

Завантажити зображення для аналізу

No file chosen

Завантажити і проаналізувати

Вставити YouTube-посилання

Завантажити і обробити

Оброблене відео з YOLO

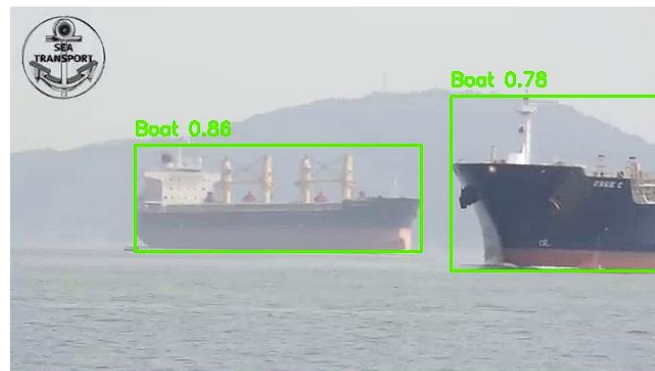


Рис 3.5.2(демонстрація аналізу статичних зображень)

Навантаження при паралельній обробці: тестування стабільності системи при одночасному надходженні декількох запитів обробки зображень і відео.

Кожен сценарій реалізовано через інтуїтивно зрозумілий веб-інтерфейс, що дозволяє користувачеві взаємодіяти з системою без необхідності мати технічні знання в галузі машинного зору. Дані про результати експериментів та продуктивність обробки наведено у наступному підрозділі.

3.5.2 Результати тестування точності розпізнавання та виявлення

Результати експериментального тестування дозволили здійснити комплексну оцінку працездатності системи моніторингу суден за ключовими критеріями: точність виявлення, швидкодія обробки, стабільність веб-сервісу та зручність користувацької взаємодії.

Точність виявлення об'єктів

Модель YOLOv11, використана у системі, продемонструвала високу якість розпізнавання об'єктів класу «судно» на різних типах зображень та відео. При середньому значенні порогу довіри $\text{conf}=0.3$ виявлялися як великі, так і дрібні об'єкти. Візуальна перевірка оброблених зображень підтвердила, що ймовірність хибно-позитивних спрацювань залишається низькою.

Швидкодія системи

Обробка одного кадру з відео займала в середньому від 40 до 90 мс, залежно від розміру зображення та кількості об'єктів на сцені. Це дозволяє забезпечити майже реальний час при потоковій обробці. Завантаження та аналіз відео з YouTube зазвичай займало 10–30 секунд (включно з завантаженням), що вважається прийнятним для асинхронної взаємодії з користувачем.

```
[ WARN:3@209.970] global cap_ffmpeg_impl.hpp:453 _opencv_ffmpeg_interrupt_
127.0.0.1 - - [15/May/2025 16:38:57] "GET /rtsp_feed HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2025 16:40:13] "GET /rtsp_feed HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2025 16:40:13] "GET /favicon.ico HTTP/1.1" 404 -
[download] 3.5% of 96.02MiB at 3.17MiB/s ETA 00:29
```

Рис3.5.3 зображення швидкості завантаження відео

Стабільність веб-додатку

У ході тестування система демонструвала стійкість до послідовної обробки декількох типів медіа. Запити на завантаження зображень, відео або запуск RTSP-потoku не спричиняли помилок у сервісі. Особлива увага приділялася унікальній генерації імен файлів відео, що дозволило уникнути конфліктів при повторному використанні однакових URL.

Користувацький інтерфейс

Інтерфейс системи виявився інтуїтивно зрозумілим: усі ключові функції доступні з головної сторінки, результати виводяться без переходу на нові сторінки, а завантаження та обробка медіа супроводжуються зрозумілим візуальним відгуком. Це суттєво спрощує використання системи для осіб без технічного досвіду.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено прототип системи моніторингу морської акваторії на основі методів глибокого навчання та комп'ютерного зору. Система використовує модель YOLOv11, яка після навчання на відповідному наборі даних продемонструвала високу точність виявлення та ідентифікації суден. Проведене експериментальне тестування показало, що система здатна обробляти відеопотік у режимі реального часу, що підтверджує її функціональність та ефективність у рамках поставлених дослідницьких задач.

Результати роботи свідчать про перспективність застосування сучасних алгоритмів візуального розпізнавання у сфері морського моніторингу. Хоча розроблена система не була інтегрована в реальні портові інфраструктури, вона може розглядатися як експериментальна основа для подальших розробок, спрямованих на підвищення автоматизації та інформатизації процесів спостереження за морськими об'єктами.

Запропоноване рішення має модульну і гнучку архітектуру, що дає можливість адаптації під різні умови експлуатації. У роботі визначено можливі напрями подальшого вдосконалення системи, зокрема розширення бази даних типів суден, оптимізація алгоритмів обробки зображень, а також інтеграція з додатковими джерелами інформації — такими як супутникові або радіолокаційні системи.

Система має потенціал застосування в наукових і освітніх цілях, а також у тестових або демонстраційних середовищах. Вона є прикладом того, як інструменти зі штучним інтелектом можуть бути використані для досліджень у галузі автоматизованого візуального спостереження за об'єктами в морському середовищі, і може слугувати основою для подальших розробок більш складних та практично орієнтованих рішень.

ПЕРЕЛІК ПОСИЛАНЬ

- 1 Європейське космічне агентство (ESA). Marine services [Електронний ресурс] // ESA. - Режим доступу: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Marine_services (дата звернення: 20.05.2025).
- 2 European Commission. Copernicus Programme. Sentinel-2 [Електронний ресурс]. - Режим доступу: <https://documentation.dataspace.copernicus.eu/Data/SentinelMissions/Sentinel2.html> (дата звернення: 21.05.2025).
- 3 Сайт Swedish Biodiversity Data Infrastructure (SBDI) iodiversitydata.se/news/sbdis-latest-tool-for-image-analysis/
- 4 Swedish Biodiversity Data Infrastructure (SBDI). SBDI's latest tool for image analysis [Електронний ресурс]. - 2024. - Режим доступу: <https://biodiversitydata.se/news/sbdis-latest-tool-for-image-analysis/> (дата звернення: 20.05.2025).
- 5 Державна екологічна інспекція України. Інформація про проведений моніторинг стану Чорного моря [Електронний ресурс]. - 2024. - Режим доступу: <https://sw.dei.gov.ua/post/informatsiya-pro-provedeniy-monitoring-stanu-chornogo-morya> (дата звернення: 20.05.2025).
- 6 National Oceanic and Atmospheric Administration (NOAA). Argo Program [Електронний ресурс] // NOAA AOML. - Режим доступу: <https://www.aoml.noaa.gov/argo/> (дата звернення: 21.05.2025).
- 7 Army Recognition. Technology: US-based Anduril launches AI-powered Seabed Sentry to transform subsea defense operations [Електронний ресурс]. - 04.04.2025. - Режим доступу: <https://www.armyrecognition.com/focus-analysis-conflicts/navy/naval-technology/technology-us-based-anduril-launches-ai-powered-seabed-sentry-to-transform-subsea-defense-operations> (дата звернення: 22.05.2025).

- 8 European Commission (iMagine project). WITOIL: Predicting Oil Spills for Cleaner Seas [Електронний ресурс]. - Режим доступу: <https://www.imagine-ai.eu/service/witoil-service> (дата звернення: 22.05.2025).
- 9 Яценко О. Уперше в Україні: акваторії морпортів моніторитимуть із космосу [Електронний ресурс] // ЕкоПолітика. - 03.02.2022. - Режим доступу: <https://ecopolitic.com.ua/ua/news/vpervye-v-ukraine-akvatorii-morportov-budut-monitorit-iz-kosmosa-2/> (дата звернення: 22.05.2025).
- 10 Naval Postgraduate School. Object Tracking and Threat Detection for Maritime Situational Awareness [Електронний ресурс]. - 2022. - Режим доступу: <https://calhoun.nps.edu/handle/10945/69627> (дата звернення: 22.05.2025).
- 11 Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). - 2016. - С. 779–788.
- 12 Wang C.-Y., Bochkovskiy A., Liao H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [Електронний ресурс] // arXiv preprint. - 2022. - Режим доступу: <https://arxiv.org/abs/2207.02696> (дата звернення: 22.05.2025).
- 13 Facebook AI Research. Detectron2 [Електронний ресурс] // GitHub. - 2024. - Режим доступу: <https://github.com/facebookresearch/detectron2> (дата звернення: 22.05.2025).
- 14 Huang J., Rathod V., Sun C. et al. Speed/accuracy trade-offs for modern convolutional object detectors // Proc. of the IEEE CVPR. - 2017. - С. 7310–7311.
- 15 Kirillov A., Mintun E., Ravi N. et al. Segment Anything [Електронний ресурс] // Meta AI Blog. - 2023. - Режим доступу: <https://segment-anything.com> (дата звернення: 22.05.2025).
- 16 Kirillov A., Mintun E., Ravi N. et al. Segment Anything [Електронний ресурс] // arXiv preprint arXiv:2304.02643. - 2023. - Режим доступу: <https://arxiv.org/abs/2304.02643> (дата звернення: 22.05.2025).

- 17 Carion N., Massa F., Synnaeve G. et al. End-to-End Object Detection with Transformers // Proc. of the European Conf. on Computer Vision (ECCV). - 2020. - Режим доступа: <https://arxiv.org/abs/2005.12872> (дата звернення: 22.05.2025)
- 18 Zhang X., Liu Y., Wang X. YOLOv11: Efficient One-stage Real-Time Object Detector for Embedded Vision [Електронний ресурс] // arXiv preprint. - 2024. - Режим доступа: <https://arxiv.org/abs/2403.00678> (дата звернення: 22.05.2025).

ПРЕЗЕНТАЦІЯ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ

**“СИСТЕМА МОНІТОРИНГУ ОБ’ЄКТІВ МОРСЬКОЇ АКВАТОРІЇ З
ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ”**

Виконав: здобувач вищої освіти групи ШД-42
Артем БЄЛЯВЦЕВ
Керівник: к.т.н. доц. Олександр ЗВЕНІГОРОДСЬКИЙ

КИЇВ - 2025

Актуальність дослідження - У сучасному світі спостерігається стрімкий розвиток цифрових технологій, що сприяє зростанню обсягу візуальної інформації. Автоматичне розпізнавання та обробка цієї інформації є важливою задачею для багатьох галузей, таких як архітектура, будівництво, інженерія, медицина та освіта.

Мета роботи - Підвищення ефективності існуючих систем управління в портах шляхом впровадження алгоритмів ідентифікації суден на основі візуального розпізнавання.

Об’єкт дослідження - Процеси інформаційні та технічні системи, що функціонують у морських портах.

Предмет дослідження - Алгоритми візуального розпізнавання суден з використанням методів штучного інтелекту.

Традиційні методи моніторингу

Супутники, AIS, патрулювання, радар

Переваги: глобальність, точність координат

Недоліки: затримки, можливість обману, низька деталізація



Вибір методів розпізнавання об'єктів

YOLOv11 – Однопрохідна модель.

- Переваги: швидкість, однопрохідна архітектура, оптимальна для потокового відео.
- Недоліки: труднощі з виявленням дрібних або частково закритих об'єктів.

SAM

- Працює без перенавчання на нових даних, підтримує інтерактивну сегментацію.
- Корисний для покращення точності аналізу, наприклад — розмірів судна.

Decetron 2

- Підходить для складних умов, де потрібна точність, а не швидкість.
- Недолік — велике навантаження на апаратне забезпечення.

Архітектура системи

- Потік відео → обробка YOLOv11 → візуалізація в браузері
- Технології: Python, Flask, OpenCV, HTML
- Результат: веб-додаток з можливістю онлайн-моніторингу

Архітектура системи

```
from ultralytics import YOLO

model = YOLO("runs/train/custom_model/weights/best.pt")

# Inference
model.predict(
    source="data.yaml",
    epochs=15,
    imgsz=640,
    batch=32,
    project="runs/train",
    name="custom_model",
    exist_ok=True
)
```

```
import cv2
import time
import numpy as np
from ultralytics import YOLO

# Load the model
model = YOLO("runs/train/custom_model/weights/best.pt")

# Load the video file
video_path = "data/video.mp4"

# Open the video file
cap = cv2.VideoCapture(video_path)

# Get the video properties
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# Create a window
cv2.namedWindow("YOLOv11 Inference")

# Main loop
while True:
    # Read a frame
    ret, frame = cap.read()

    # If frame is read correctly (ret == True)
    if ret:
        # Run inference
        results = model.predict(frame)

        # Show results
        for result in results:
            boxes = result.boxes

            for box in boxes:
                x1, y1, x2, y2 = box.xyxy[0]
                x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

                # Draw a rectangle
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

                # Get the class name and confidence
                class_name = result.names[box.cls]
                conf = result.conf[0]

                # Draw the label
                cv2.putText(frame, f"{class_name} {conf:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 1)

        # Show the frame
        cv2.imshow("YOLOv11 Inference", frame)

        # Wait for a key press
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    else:
        break

# Release the video capture object and close the window
cap.release()
cv2.destroyAllWindows()
```

Архітектура системи

```
def normalize_and_resize(frame, width=640):
    normalized = cv2.normalize(frame, None, 0/255, 255/255, cv2.NORM_MINMAX)
    height = int(frame.shape[0] * (width / frame.shape[1]))
    resized = cv2.resize(normalized, (width, height))
    return resized

def denoise_frame(frame):
    return cv2.fastNlMeansDenoising(frame, None, 0.1, 10, 3)

def correct_lighting(frame):
    lab = cv2.cvtColor(frame, cv2.COLOR_LAB2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    l_clahe = clahe.apply(l)
    lab_sq = cv2.merge([l_clahe, a, b])
    corrected = cv2.cvtColor(lab_sq, cv2.COLOR_LAB2BGR)
    return corrected

# --- Show the results of the steps ---
def process_frame(frame):
    frame = normalize_and_resize(frame)
    frame = denoise_frame(frame)
    frame = correct_lighting(frame)
    results = model.predict(frame, verbose=1, confidence=0.5)
```

```
def gen_frames(path):
    cap = cv2.VideoCapture(path)
    while True:
        success, frame = cap.read()
        if not success:
            break
        annotated = process_frame(frame)
        ret, buffer = cv2.imencode('.jpg', annotated)
        frame_bytes = buffer.tobytes()
        yield (b'--frame\r\n' + frame_bytes + b'\r\n')
        cap.release()

@app.route('/video_feed')
def video_feed():
    filename = session.get('last_video', None)
    if filename:
        path = os.path.join('static', filename)
        return Response(gen_frames(path), mimetype='multipart/x-mixed-replace; boundary=frame')
    else:
        return "Error: no video", 404

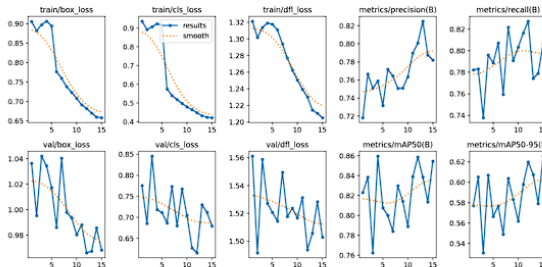
@app.route('/')
def index():
    last_video = session.get('last_video', None)
    return render_template("index.html", last_video=last_video)

@app.route('/youtube_process', methods=['POST'])
def youtube_process():
    youtube_url = request.form['youtube_url']
    filename = download_youtube_video(youtube_url)
    session['last_video'] = filename
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Результати навчання

- Досягнута точність розпізнавання – до 89%
- Найкраще виявляються великі судна
- Складності – з малими або закритими об'єктами



Інтерфейс користувача

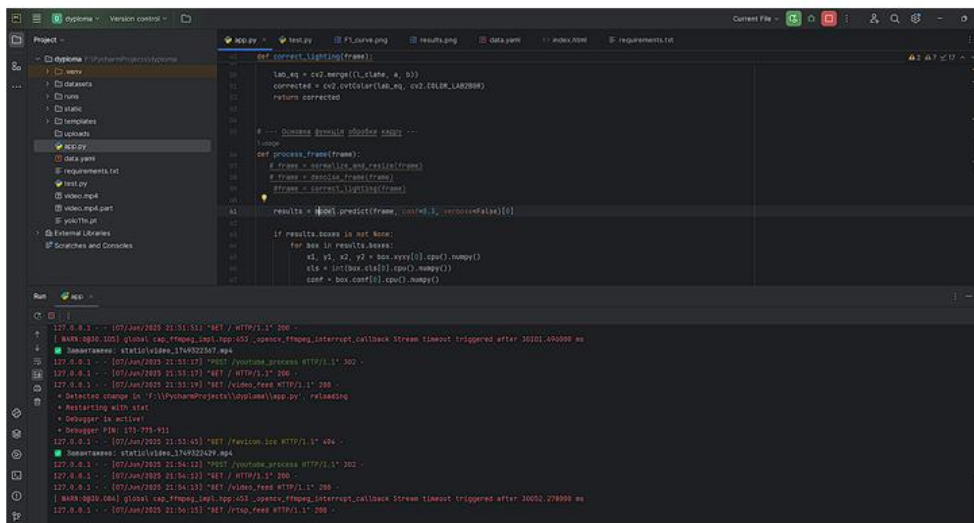
Система моніторингу суден

Завантажити зображення для аналізу

Вставити YouTube-посилання

Оброблене відео з YOLO

Демонстрація роботи



Висновки

- Розроблено систему виявлення суден у режимі реального часу
- Досягнуто баланс точності та швидкодії
- Система придатна до розширення та адаптації до портових умов

ДОДАТОК А

Основний модуль

```
import os
import cv2
import time
import numpy as np
from flask import Flask, Response, render_template, request, redirect, url_for,
session
from ultralytics import YOLO
import yt_dlp

app = Flask(__name__)
app.secret_key = "your_secret_key" # необхідно для session

model = YOLO("runs/train/custom_model8/weights/best.pt")

def download_youtube_video(url):
    # Генеруємо унікальне ім'я файлу
    timestamp = int(time.time())
    filename = f"video_{timestamp}.mp4"
    output_path = os.path.join("static", filename)

    ydl_opts = {
        'format': 'best[ext=mp4]',
        'outtmpl': output_path,
        'quiet': True,
    }

    with yt_dlp.YoutubeDL(ydl_opts) as ydl:
        ydl.download([url])

    print(f"☑ Завантажено: {output_path}")
    return filename

# --- Обробка кадру: препроцесінг ---
def normalize_and_resize(frame, width=640):
    normalized = cv2.normalize(frame, None, 0, 255, cv2.NORM_MINMAX)
```

```

height = int(frame.shape[0] * (width / frame.shape[1]))
resized = cv2.resize(normalized, (width, height))
return resized

def denoise_frame(frame):
    return cv2.bilateralFilter(frame, d=9, sigmaColor=75, sigmaSpace=75)

def correct_lighting(frame):
    lab = cv2.cvtColor(frame, cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab)

    clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
    l_clahe = clahe.apply(l)

    lab_eq = cv2.merge((l_clahe, a, b))
    corrected = cv2.cvtColor(lab_eq, cv2.COLOR_LAB2BGR)
    return corrected

# --- Основна функція обробки кадру ---
def process_frame(frame):
    # frame = normalize_and_resize(frame)
    # frame = denoise_frame(frame)
    # frame = correct_lighting(frame)

    results = model.predict(frame, conf=0.3, verbose=False)[0]

    if results.bboxes is not None:
        for box in results.bboxes:
            x1, y1, x2, y2 = box.xyxy[0].cpu().numpy()
            cls = int(box.cls[0].cpu().numpy())
            conf = box.conf[0].cpu().numpy()
            label = f"{results.names[cls]} {conf:.2f}"
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0),
2)

            cv2.putText(frame, label, (int(x1), int(y1) - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)

    return frame

def gen_frames(path):
    cap = cv2.VideoCapture(path)

```

```

while True:
    success, frame = cap.read()
    if not success:
        break
    annotated = process_frame(frame)
    ret, buffer = cv2.imencode('.jpg', annotated)
    frame_bytes = buffer.tobytes()
    yield (b'--frame\r\n'
           b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n')
cap.release()

@app.route('/video_feed')
def video_feed():
    filename = session.get("last_video", None)
    if filename:
        path = os.path.join("static", filename)
        return Response(gen_frames(path), mimetype='multipart/x-mixed-replace;
boundary=frame')
    else:
        return "Відео не знайдено", 404

def generate_rtsp():
    cap =
cv2.VideoCapture("rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny_115k.mov")
    while True:
        success, frame = cap.read()
        if not success:
            break
        ret, buffer = cv2.imencode('.jpg', frame)
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

@app.route('/rtsp_feed')
def rtsp_feed():
    return Response(generate_rtsp(), mimetype='multipart/x-mixed-replace;
boundary=frame')

```

```

@app.route('/')
def index():
    last_video = session.get("last_video", None)
    return render_template('index.html', last_video=last_video)

@app.route('/youtube_process', methods=['POST'])
def youtube_process():
    youtube_url = request.form['youtube_url']
    filename = download_youtube_video(youtube_url)
    session["last_video"] = filename
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```

Функція навчання моделі на основі датасету

```

from ultralytics import YOLO

model = YOLO("runs/train/custom_model8/weights/best.pt")

# Тренування
model.train(
    data="data.yaml",
    epochs=15,
    imgsz=640,
    batch=16,
    project="runs/train",
    name="custom_model",
    resume=False
)

```