

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичного моделювання та аналізу даних**

«До захисту допущено»

В.о. завідувача кафедри

Г.О.ЯЙЛИМОВА

« » _____ 2024 р.

Дипломна робота
на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика
на тему: «**Моделі семантичної сегментації в задачі пошуку
нафтових плям на основі геопросторових даних**»

Виконав:

студент 4 курсу, групи ФІ-01

Григоренко Є.М. _____

Керівник:

Д.т.н., проф. Шелестов А.Ю. _____

Рецензент:

К.т.н, старший дослідник, Яйлимов Б.Я. _____

Засвідчую, що у цій
дипломній роботі немає
запозичень з праць інших
авторів без відповідних
посилань.

Студент

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичного моделювання та аналізу даних**

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи моделювання, розпізнавання образів та
комп'ютерного зору»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Г.О. ЯЙЛИМОВА

«__» _____ 2023 р.

**ЗАВДАННЯ
на дипломну роботу**

Студент: Григоренко Євгеній Михайлович

1. Тема роботи: *«Моделі семантичної сегментації в задачі пошуку нафтових плям на основі геопросторових даних»*,

керівник: д.т.н., професор Шелестов А.Ю.,

затверджені наказом по університету №2251-С від «31» травня 2024 р.

2. Термін подання студентом роботи: «7» червня 2024 р.

3. Об'єкт дослідження: Джерела даних та моделі семантичної сегментації, які використовуються для оцінки забруднення морів.

4. Предмет дослідження: Методи збору та аналізу, які використовуються для оцінки забруднення морів нафтовими плямами, зокрема геопросторові дані та сучасні моделі семантичної сегментації.

5. Перелік завдань: Дослідити предметну область, здійснити огляд літератури та ресурсів, присвячених забрудненню морів, визначити основні джерела даних та основні платформи, що надають дані про стан морського середовища. Проаналізувати технології, що використовуються

для супутникових спостережень, визначити їх переваги та обмеження. Підготувати дані і створити датасет із різних джерел супутникових даних, провести експеримент і проаналізувати метрики і результати роботи різних архітектур моделей сегментації.

6. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): *Презентація доповіді*

7. Дата видачі завдання: 20 квітня 2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-20 квітня 2024 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	20 квітня - 2 травня 2024 р.	Виконано
3	Аналіз методів та технологій збору даних, їх переваг та обмежень	2-8 травня 2024 р.	Виконано
4	Огляд та аналіз сучасних архітектур моделей семантичної сегментації та функцій втрат	20 квітня-15 травня 2024 р.	Виконано
5	Створення датасету і проведення експерименту	8-20 травня 2024 р.	Виконано
6	Аналіз отриманих під час експерименту результатів, оформлення дипломної роботи	20 травня-15 червня 2024 р.	Виконано

Студент

_____ Григоренко Є.М.

Керівник

_____ Шелестов А.Ю.

РЕФЕРАТ

Кваліфікаційна робота містить: 57 стор., 14 рисунків, 3 таблиці, 23 джерел.

Ця робота є актуальною через зростаючу потребу виявлення нафтових плям у водах для моніторингу морського середовища та прийняття заходів щодо їх ліквідації. Сучасні методи семантичної сегментації зображень, засновані на супутникових даних, дозволяють значно підвищити ефективність виявлення та контролю забруднень. У роботі використовуються дані звітів про спостереження за забрудненням моря (MPSR) для створення датасету та супутникові знімки для аналізу та експериментів.

Робота складається з декількох етапів, включаючи огляд літератури, підготовку датасету, імплементацію моделей сегментації та оцінку їх продуктивності. Для реалізації та навчання моделей було використано бібліотеку PyTorch Lightning. Особлива увага приділена порівнянню різних архітектур моделей та функцій втрат для семантичної сегментації нафтових плям. У ході численних експериментів оцінено точність моделей за метриками Intersection over Union (IoU) та F1-Score (Dice Score). Результати показали, що використання таких архітектур моделей як UNet++ та DeepLabV3+, забезпечує високу точність виявлення нафтових плям. Дослідження також виявило оптимальні параметри навчання для цих моделей, що може бути використано для подальшого моніторингу забруднень морського середовища.

Таким чином, результати даного дослідження дозволяють визначити ефективні моделі та параметри для семантичної сегментації нафтових плям у водах на основі супутникових зображень, що сприяє покращенню методів моніторингу та охорони довкілля.

КЛЮЧОВІ СЛОВА: СЕМАНТИЧНА СЕГМЕНТАЦІЯ, СУПУТНИКОВІ ДАНІ, ВИЯВЛЕННЯ НАФТОВИХ ПЛЯМ, UNET++, DEEPLABV3+

ABSTRACT

The qualifying paper contains: 57 pages, 14 figures, 3 tables, 23 sources.

This work is relevant due to the growing need for detecting oil spills in waters for marine environment monitoring and response measures. Modern methods of semantic image segmentation based on satellite data significantly enhance the effectiveness of pollution detection and control. The study uses data from Marine Pollution Surveillance Reports (MPSR) and satellite imagery for analysis and experiments.

The research includes several stages, such as a literature review, dataset preparation, model implementation, and performance evaluation. The PyTorch Lightning library was used for model implementation and training. Special attention is paid to comparing different model architectures and loss functions for oil spill semantic segmentation. Extensive experiments were conducted to evaluate the accuracy of the models using metrics like Intersection over Union (IoU) and F1-Score (Dice Score). The results showed that using specific model architectures, such as UNet++ and DeepLabV3+, provides high accuracy in detecting oil spills. The study also identified optimal training parameters for these models, which can be used for further marine pollution monitoring.

Thus, the findings of this research help identify effective models and parameters for semantic segmentation of oil spills in waters based on satellite imagery, contributing to improved environmental monitoring and protection methods.

KEYWORDS: SEMANTIC SEGMENTATION, SATELLITE DATA, OIL SPILL DETECTION, UNET++, DEEPLABV3+

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1 ОГЛЯД ЛІТЕРАТУРИ І ДОСЛІДЖЕНЬ НА ТЕМУ ОЦІНКИ ЗАБРУДНЕННЯ МОРІВ	11
Висновки по розділу 1	15
2 ТЕХНОЛОГІЇ СПОСТЕРЕЖЕННЯ ЗЕМЛІ ДЛЯ ОЦІНКИ ЯКОСТІ МОРСЬКОГО СЕРЕДОВИЩА	17
2.1 Типи сенсорів	17
2.2 Огляд супутників та їх характеристик	21
2.2.1 Характеристики супутника Sentinel-1.....	21
2.2.2 Характеристики супутника Sentinel-2.....	24
Висновки по розділу 2	25
3 ВИКОРИСТАНІ ДАНІ.....	26
Висновки по розділу 3	30
4 ПОБУДОВА МОДЕЛЕЙ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ, ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ	32
4.1 Функції втрат.....	32
4.2 Вибір моделей та параметрів.....	34
4.3 Аналіз результатів роботи моделей.....	38
Висновки по розділу 4	42
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44
ДОДАТОК А ТЕКСТИ ПРОГРАМ	47
A1 Завантаження даних Sentinel-1	47
A2 Завантаження даних Sentinel-2.....	51

А3 Створення вбох.....	54
А4 Створення композиту Oil Spill Index.....	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EO - Ecological Objective

IMAP - Integrated Monitoring and Assessment Programme

SAR - Synthetic-aperture radar

NIR - Near-infrared

NDWI - Normalized Difference Water Index

NDVI - Normalized difference vegetation index

SWM - Sentinel Water Mask

NDSII - Normalized Difference Snow/Ice Index

SVM - Support vector machine

CNN - Convolutional neural network

DBN - Deep Belief Network

TIR - Thermal Infrared

SWIR - Shortwave Infrared

ASF - Alaska Satellite Facility

AOI – Area of interest

ВСТУП

Забруднення морського середовища нафтовими плямами становить серйозну загрозу для екосистем, економіки та здоров'я людей. Нафтові плями можуть виникати в результаті аварій нафтовозів, незаконного скидання нафти або відсутності відповідних заходів управління нафтопроводами. Інциденти такого роду супроводжуються рядом наслідків, включаючи негативний вплив на морське біорізноманіття, рибальство, економіку прибережних країн та здоров'я людей, що залежать від морських ресурсів. Також, одним із ключових викликів є діяльність так званого тіньового флоту — суден, що оперують поза законодавчими рамками, уникаючи виявлення та відстеження. Ці судна часто використовуються для незаконного перевезення нафти, в результаті чого можуть виникати аварії і викликати серйозне забруднення морів. Використання супутникових знімків для виявлення нафтових плям є ефективним інструментом у боротьбі з тіньовим флотом. Сучасні технології дозволяють здійснювати моніторинг великих акваторій, виявляючи навіть невеликі нафтові розливи. Різні методи аналізу супутникових знімків дозволяють автоматично виділяти нафтові плями та ідентифікувати джерело забруднення. Це важливо для оперативного втручання та негайного виявлення винних суб'єктів. Таким чином, подальше дослідження та розробка ефективних методів виявлення нафтових плям є важливими етапами для забезпечення екологічної безпеки морських вод. Вони не тільки сприяють оперативному реагуванню на надзвичайні ситуації, але й допомагають виявляти та зменшувати незаконну діяльність тіньового флоту.

Актуальність дослідження полягає у важливості проблеми забруднення морів як серйозної екологічної проблеми, яка значно впливає на морські екосистеми. Доступ до надійних джерел даних та алгоритмів для оцінки забруднення морів є вирішальним для дослідження стану

середовища, ідентифікації забруднювачів, визначення їх походження та запобігання подальшому поширенню. Мета дослідження полягає у проведенні аналізу різноманітних джерел даних та моделей, що використовуються для оцінки забруднення морів. Визначення характеристик, переваг та обмежень цих джерел, таких як супутникові спостереження, дані з морських станцій та моніторингові програми, є важливою частиною дослідження. Завдання дослідження включає збір та систематизацію інформації про джерела даних для оцінки забруднення морів, аналіз технологій та методів їх збору та обробки, а також визначення метрик надійності та точності кожного джерела даних або продукту. Об'єкт дослідження становлять джерела даних та моделі семантичної сегментації, які використовуються для оцінки забруднення морів, тоді як предмет дослідження включає методи їх збору, аналізу та їх ефективність. Практичне застосування отриманих результатів дослідження може полягати у поліпшенні моніторингу та оцінки забруднення морських вод. Отримані дані та аналіз різних джерел можуть бути використані владними органами, науковими установами та організаціями, що займаються охороною довкілля для прийняття ефективних рішень. Наукова новизна дослідження полягає у розробці ефективних моделей для семантичної сегментації, що дозволяють аналізувати та виявляти нафтові розливи з високою точністю.

1 ОГЛЯД ЛІТЕРАТУРИ І ДОСЛІДЖЕНЬ НА ТЕМУ ОЦІНКИ ЗАБРУДНЕННЯ МОРІВ

Проблема забруднення морів є серйозною екологічною проблемою, яка має значний вплив на морські екосистеми. Важливо мати доступ до надійних джерел даних та продуктів для оцінки забруднення морів, що допоможе вивчити стан середовища, ідентифікувати забруднювачі, визначити їх походження та не допустити їх поширення. Зменшення ризику нафтових розливів є надзвичайно важливим для захисту навколишнього середовища та зменшення економічних втрат. Спостереження за нафтовими розливами - важлива складова управління нафтовими катастрофами.

Прогрес у технологіях дистанційного зондування може допомогти встановити причину та винуватців забруднення та виявити незначні розливи до того, як вони спричинять широкомасштабні пошкодження. Результати дослідження можуть бути використані для покращення моніторингу та оцінки забруднення морських вод. Отримані дані та аналіз різних джерел можуть бути використані органами влади, дослідницькими установами та організаціями, що займаються охороною довкілля, для прийняття рішень щодо зменшення забруднення морів та захисту морських екосистем.

Перед тим, як перейти до конкретних типів забруднень і їх вивчення, спочатку дослідили доповіді екологічних організацій і державних установ щодо моніторингу та контролю різних параметрів морських середовищ, визначили основні типи індикаторів та параметрів, за якими оцінюють якість морської води і екосистеми.

У доповіді [1] представлено перелік екологічних цілей (ЕО) та індикаторів, які використовуються в рамках міжнародної програми ІМАР для моніторингу якості морських середовищ. Ці показники охоплюють широкий спектр екологічних аспектів, починаючи від біорізноманіття та некорінних видів до забруднення, морського сміття та підводного шуму. Кожна екологічна ціль пов'язана з конкретними індикаторами, які забезпечують вимірювані параметри для оцінки стану та тенденцій морського середовища. ЕО9 (забруднення) включає такі показники, як концентрація ключових шкідливих забруднюючих речовин. Надалі

ми будемо концентруватись детальніше на конкретній екологічній цілі – ЕО9, тобто хімічне забруднення морського середовища, зокрема нафтові розливи.

В статті [2] автори надають огляд методів дистанційного зондування, поєднання фізичних та оптичних датчиків, використання біосенсорів та даних, які використовуються для моніторингу морських середовищ, зокрема забруднення твердими частками, нафтою, мікропластиком та іншими хімічними речовинами.

В посібнику [3] автори узагальнюють і підсумовують інформацію, представлену у звітах Американського інституту нафти та екологічних організацій і детально описують технології дистанційного моніторингу та їх обмеження.

Автори статті [4] розглядають задачу моніторингу забруднення морського середовища, ідея якої полягає в застосуванні технології великих даних для того, щоб поліпшити методи визначення масштабів забруднення а також пропонують метод моніторингу, за допомогою якого можна з високою точністю та швидкістю ідентифікувати основну інформацію про забруднюючі речовини.

Наступні джерела літератури стосуються класифікації, сегментації та оцінки товщини нафтового покриву.

У статті [5] автори проводять дослідження вилливу нафти з використанням алгоритмів класифікації зображень з супутників Landsat 5, Landsat 7, Landsat 8, Sentinel-1, Sentinel-2. Результати класифікації плям нафти в технологіях активного та пасивного виявлення розходяться, відповідно це впливає на методологію вибору методів обробки в різних ситуаціях і областях. Також автори пропонують порівняння знімків з різних дат (для однієї і тієї ж території) для вирахування змін у воді і використання індексів на додаток до RGB та NIR діапазонів: NDVI для вивчення рослинності, NDWI2 та SWM для моніторингу води у водосховищах та NDSII для оцінки впливу снігового покриву. У статті [6] автор об'єднує глибоке та поверхневе навчання, досліджує факт того, що зображення нафтового забруднення відображають різні ознаки на різних масштабах, на основі багатомасштабних ознак після вейвлет-перетворення використовує два класичних алгоритми поверхневого навчання та два методи глибокого навчання для виявлення забруднення. Також, застосовує запропонований метод до гіперспектрального зображення з БПЛА і

порівнює результати запропонованого методу з іншими популярними методами такими як метод опорних векторів (SVM), DBN, 1-D згорткова нейронна мережа(1D-CNN) та MRF-CNN. Стаття [7] була проаналізована з метою визначити, чи можна за допомогою гіперспектральних сенсорів ідентифікувати і розрізнити нафтові емульсії і різні види нафти.

Визначили, що нафтові емульсії можуть бути чітко ідентифіковані, а різні види пролітої нафти і нафтові плівки можуть бути визначені на основі різниці спектральних діапазонів. В статті автор використовує дискретну відстань Фреше для оцінки схожості між спектральними кривими.

В статті [8] автор використовує глибокі згорткові нейронні мережі (DCNN) для вивчення різних фізичних та контекстуальних ознак для сегментації а також архітектурою кодер-декодер з використанням попередньо навчених CNN таких як UNet, LinkNet, PSPNet, DeepLabv2, DeepLabv2 (msc), DeepLabv3+ і визначенням таких об'єктів як море, нафтове забруднення, явища подібні нафтовому забрудненню, кораблів, суходолу(табл 1.1).

Таблиця 1.1 - Порівняння моделей, IOU(%) [8].

Model	Sea Surface	Oil Spill	Look-alike	Ship	Land	mIoU
UNet	93.90	53.79	39.55	44.93	92.68	64.97
LinkNet	94.99	51.53	43.24	40.23	93.97	64.79
PSPNet	92.78	40.10	33.79	24.42	86.90	55.60
DeepLabv2	94.09	25.57	40.3	11.41	74.99	49.27
DeepLabv2(msc)	95.39	49.53	49.28	31.26	88.65	62.83
DeepLabv3+	96.43	53.38	55.40	27.63	92.44	65.06

На відміну від класифікації кожного пікселя за допомогою AlexNet на наявність або відсутність нафтового розливу, автор статті [9] розділяє початкове зображення SAR з супутників Gaofen-3 і Radarsat-2 на декілька менших зображень з деяким перетином. Подальша класифікація відбувається для кожного зображення

на наявність або відсутність нафтового розливу. Якщо більше 60% пікселів у маленькому зображенні класифікуються як нафтовий розлив, то вважається, що все зображення містить нафтовий розлив. Такий підхід надає більш цілісну оцінку наявності нафтових розливів.

В статті [10] автор досліджує методи перетворення зображень з супутнику Sentinel-2. З цієї статті можна зробити висновок, що ділення одного спектрального каналу на інший створює зображення, яке надає відносні інтенсивності каналів, відповідно спектральні різниці між каналами дозволяють краще інтерпретувати особливості зображення.

Автори статті [16] пропонують новий метод аугментації даних з використанням генеративних протистоянь нейронних мереж (GAN) з перетворенням піксель-до-пікселя (pix2pix). Цей підхід генерує реалістичні синтетичні зображення супутників з штучними масками правдивості, навіть для рідкісних розподілів класів сільськогосподарських культур. Він дозволяє створювати додаткові вибірки менших класів, покращуючи баланс тренувальних даних і перевершуючи традиційні методи аугментації.

У статті [18] автори розглядають основні (з погляду геопросторового аналізу) аспекти глибокого навчання, а саме: підвищення роздільної здатності графічних даних, використання transfer learning для оптимізації процесів навчання, масштабування моделей глибоких нейронних мереж та аналіз часових рядів за допомогою рекурентних нейронних мереж.

Дослідження [20], [22], [23] зосереджуються на оцінці параметрів якості води у Середземному морі з акцентом на концентрацію хлорофілу-а. Використовуючи transfer learning, автори адаптують модель, спочатку навчену на супутникових знімках Sentinel-2 та даних з місцевих станцій контролю концентрації хлорофілу в Гонконзі, для застосування в південному узбережжі біля Лімасолу, Кіпр, у рамках проекту iMERMAID Horizon Europe.

В роботах [19], [21] автори впроваджують підхід transfer learning з використанням мережі LinkNet для семантичної сегментації зображень SAR в регіоні Середземного моря, зокрема біля Кіпру, в рамках проекту iMERMAID.

Horizon Europe. Автори використовують набір даних Sentinel-1 SAR для тренування адаптивної моделі для виявлення нафтових забруднень. Результати показують F1-score 0.721 та IoU 0.564 на валідаційному наборі, що підтверджує ефективність запропонованого підходу в областях Атлантичного океану та Середземного моря.

У зв'язку з великою кількістю способів і засобів, сенсорів, що використовуються для спостереження за нафтовими розливами, потрібен всебічний огляд та порівняння існуючих технологій. Зокрема, треба проаналізувати характеристики та застосування різних датчиків, адже краще розуміння їх переваг та недоліків сприятиме більш ефективному використанню для реагування на нафтові розливи та планування заходів у разі надзвичайних ситуацій. На жаль, жоден окремий датчик і сенсор не може надати всю інформацію, яка необхідна для планування заходів у разі нафтового розливу, тому що більшість з них мають певні обмеження в низькій розмірності, невеликій області застосування тощо. Тобто існуючі методи мають обмежений спектр фізичних чи хімічних характеристик, які можна виявити, оцінити або виміряти. Тому в даний час для цього використовуються комбінації сенсорів та датчиків. Зокрема, супутникові сенсори використовуються для попередньої оцінки нафтового розливу, тоді як повітряні використовуються для детального аналізу. Хоча супутникове дистанційне зондування не підходить для тактичного планування заходів, воно може забезпечити швидке та оглядове охоплення пошкодженої території.

Висновки по розділу 1

У розділі 1 було розглянуто та проаналізовано літературу про сучасні методи оцінки забруднення морів, зокрема нафтових розливів, які є серйозною екологічною загрозою для морських екосистем. Важливо мати доступ до надійних даних і продуктів для ефективного моніторингу стану морських середовищ, ідентифікації джерел забруднення та розробки стратегій зменшення їх впливу.

Основні аспекти, які були проаналізовані, включають значення технологій дистанційного зондування для виявлення нафтових розливів. Супутникові спостереження та гіперспектральний аналіз є ключовими інструментами для оперативного виявлення та моніторингу нафтових забруднень на великих територіях. Також була досліджена література щодо використання сучасних моделей машинного навчання, таких як UNet, LinkNet, PSPNet і DeepLab для сегментації супутникових зображень і виявлення на них нафтових плям. Крім того, наголошено на потребі інтеграції різних типів сенсорів і датчиків для комплексного моніторингу. Такий підхід дозволяє не тільки точно ідентифікувати забруднення, а й ефективно планувати відповідні заходи.

2 ТЕХНОЛОГІЇ СПОСТЕРЕЖЕННЯ ЗЕМЛІ ДЛЯ ОЦІНКИ ЯКОСТІ МОРСЬКОГО СЕРЕДОВИЩА

Сучасні технології спостереження Землі відіграють ключову роль у моніторингу морських екосистем і оцінці їхнього стану. Використання технологій дистанційного зондування дозволяє отримувати деталізовані дані про різноманітні аспекти морського середовища, включаючи забруднення нафтою, стан морських поверхонь і вплив кліматичних змін. У цьому розділі ми розглянемо різні типи супутників та встановлених на них сенсорів, їхні характеристики і особливості застосування для ефективного моніторингу морської поверхні.

2.1 Типи сенсорів

Сенсори на супутниках створюють зображення, використовуючи різні довжини хвиль в межах електромагнітного спектру. Спочатку ці сенсори вимірюють різні властивості, такі як відбиття, поглинання або випромінювання від об'єктів на поверхні Землі або в морському середовищі. Отримані вимірювання фіксуються у вигляді піксельних даних, які потім об'єднуються для створення повних зображень.

Залежно від довжини хвилі, яку використовує сенсор, вимірювання можуть бути активними або пасивними. Активні сенсори випромінюють світло чи радіохвилю, а потім реєструють сигнал, який повертається назад (наприклад, радари). Пасивні сенсори реєструють електромагнітне випромінювання, яке є відбиттям або розсіянням від об'єктів.

Ці технології дозволяють отримувати різні типи зображень, які можуть бути використані для аналізу різних аспектів земної поверхні та морського середовища, включаючи дослідження кліматичних змін, контроль за екологічними забрудненнями та моніторинг природних ресурсів. Відмінності між активними та пасивними сенсорами показано в табл. 2.1.

Таблиця 2.1 - Відмінності активних та пасивних сенсорів

	Пасивні сенсори	Активні сенсори
Процес	Пасивні датчики виявляють природне випромінювання, що випромінюється або відбивається спостережуваними цілями або явищами	Активні датчики випромінюють власний імпульс енергії, а потім вимірюють сигнал, який відбивається назад до датчика.
Переваги	Видимі зображення зазвичай легко обробити та інтерпретувати.	Здатні працювати вдень або вночі, а також за деяких несприятливих умов. Можуть бути налаштовані для оптимізації вибірки поверхні і фокусування енергії для досягнення високої просторової роздільної здатності або для мінімізації атмосферного поглинання.
Обмеження	Мають обмеження щодо часу використання, наприклад, вдень і за ясної погоди. Можуть залежати від погодних умов, наприклад, випромінювання може поглинатися або спотворюватися хмарами, туманом тощо.	Більш складні, ніж пасивні датчики, з точки зору технології, і створюють складнощі та проблеми в обробці та інтерпретації зображень.

Найпоширенішими типами пасивних сенсорів є ті, що працюють у видимому, інфрачервоному та тепловому інфрачервоному діапазонах, їх також називають оптичними. Найпоширенішим типом активного сенсора є радар із синтетичною апертурою(SAR), який використовує мікрохвильовий діапазон.

Особливості та обмеження різних типів датчиків для виявлення нафти:

Радар із синтезованою апертурою (SAR):

- Працює як вдень, так і вночі.
- Ефективний при хмарному небі.
- Може працювати в умовах хмар, дощу або туману.
- Погано працює в дуже спокійних умовах моря (швидкість вітру менше 3 м/с) або при сильних хвилях (швидкість вітру більше 12 м/с).

- Погано працює у відкритій воді з концентрацією льоду більше 30% або на льоду.

- Тепловий інфрачервоний (TIR): Працює як вдень, так і вночі за умови ясного неба.

- Не працює в несприятливих погодних умовах.

Короткохвильовий інфрачервоний (SWIR) / ближній інфрачервоний (NIR):

- Працює тільки у денний час і під час сутінків (SWIR).

- Ефективний у денний час (NIR).

- Працює тільки в ясну погоду з обмеженою хмарністю.

- SWIR може працювати в умовах туману.

Видиме світло (VIS):

- Працює тільки у денний час.

- Не працює в несприятливих погодних умовах.

Обмеження для всіх типів датчиків:

- Усі ці датчики не дуже добре працюють в грубих умовах моря.

- SWIR і TIR не працюють у несприятливих погодних умовах.

- Датчики TIR, SWIR, NIR і VIS неефективні в умовах сильного хвилювання моря.

- У більшості льодових середовищ ці датчики працюють, за винятком випадків, коли нафта інкапсульована в льоду або під ним.

Кожен тип сенсору використовує різні області різних довжин хвиль в електромагнітному спектрі(табл. 2.2).

Таблиця 2.2 - Типи сенсорів і їх діапазони

Тип сенсора	Діапазон сенсора	Електромагнітний діапазон	Що вимірює
Оптичний	Видимий	0.4–0.7 мкм	Відбите сонячне світло
	Інфрачервоний	Ближній = 0,74–1,4 мкм Короткохвильовий = 1,4–3,0 мкм	Природне випромінювання
	Тепловий інфрачервоний	3.0–14.0 мкм	Температура поверхні
SAR	Мікрохвильовий	X-діапазон = 8–12 ГГц / 2.5–3.75 см C-діапазон = 4–8 ГГц / 3.75–7.5 см L-діапазон = 1–2 ГГц / 15–30 см P-діапазон = 0.2998–0.999 ГГц / 30–100 см	Зворотне розсіювання радару

Оскільки у нас тепер є кілька спектральних діапазонів, кожен з яких фіксує різні довжини хвиль, ми можемо поєднувати різні діапазони, щоб створювати нові композитні зображення, які можуть показувати різні характеристики нашої області. Через використання кількох спектральних діапазонів цей тип супутникових зображень також називається мультиспектральним зображенням (MSI).

Багатоспектральні сенсори широкого діапазону спроектовані для використання відносно невеликої кількості дискретних діапазонів (від 4 до 50) у межах певних довжин хвиль. Гіперспектральні сенсори використовують більше 100 діапазонів на різних оптичних довжинах хвиль (включаючи тепловий інфрачервоний діапазон). Їх просторове розрізнення є відносно малим, однак вони забезпечують широкий спектр частот, що може бути корисним для виявлення та, в деяких випадках, оцінки характеристик вилливу нафти.

Високороздільні мультиспектральні сенсори можуть досягати вищої просторової роздільності, але матимуть обмежену спектральну дискретизацію лише у видимих та близьких інфрачервоних діапазонах; у результаті вони ефективні лише при використанні у денний час за умов відсутності хмар.

Збір даних у кількох спектральних діапазонах може зменшити ймовірність

хибних спрацьовувань; однак, обсяг зібраних даних є значним, і обробка може бути тривалою, що призводить до збільшення часу обробки даних, а також вимагає більших технічних та обчислювальних ресурсів.

Окрім мультиспектральних сенсорів, доступні мультисенсорні супутники, які поєднують кілька різних сенсорів, таких як ультрафіолетові та теплові інфрачервоні, для покращення виявлення нафти та зменшення ймовірності хибних спрацьовувань. Однак ці комбінації сенсорів все одно підлягають тим самим обмеженням, що й окремі сенсори.

Оскільки зазвичай використовувані моделі машинного навчання вимагають трьохвимірних зображень, ми можемо поєднати кілька діапазонів в один діапазон, виконуючи операції над елементами на кількох діапазонах.

2.2 Огляд супутників та їх характеристик

Сучасні супутники грають ключову роль у глобальному моніторингу Землі, надаючи важливі дані для різноманітних досліджень та застосувань. Одними з найбільш застосованих супутникових систем є Sentinel-1 та Sentinel-2, які є частиною програми Copernicus та забезпечують доступ до безкоштовних даних високої якості з різноманітними характеристиками і продуктами обробки.

2.2.1 Характеристики супутника Sentinel-1

Місія Sentinel-1 це це Європейська Радіолокаційна Обсерваторія в рамках спільної ініціативи Copernicus Європейської Комісії (ЕС) та Європейського космічного агентства (ESA). Copernicus - це європейська ініціатива з впровадження інформаційних послуг, пов'язаних з довкіллям та безпекою. Вона базується на даних спостережень, отриманих від супутників спостереження Землі та наземної інформації.

Місія Sentinel-1 складається з двох сонячно-синхронних супутників на полярних орбітах, які поділяють одну орбітальну площину з різницею в фазі 180°, працюють вдень і вночі, і виконують зйомку за допомогою С-діапазону синтетичної апертурної радіолокації, що дозволяє їм отримувати зображення

незалежно від погодних умов. Вона працює в чотирьох унікальних режимах зйомки з різною роздільною здатністю (до 5 м) і покриттям (до 400 км). Вона забезпечує можливість двополяризаційної зйомки та дуже короткі інтервали повторних відвідувань. Для кожного спостереження доступні точні вимірювання положення та орієнтації космічного апарату. Вимірювальні можливості цієї місії охоплюють топографію ландшафту, багатофункціональні зображення (суша), багатофункціональні зображення (океан), поверхневі вітри океану, топографію/течії океану, висоту і спектр хвиль океану, покриття морським льодом, край і товщину, покриття снігом, край і глибину.

Sentinel-1 працює в чотирьох режимах зйомки: Stripmap(SM), Interferometric Wide Swath(IW), Extra-Wide Swath(EW) та Wave(WV) (рис 2.1). Режими зйомки SM, IW та EW можуть працювати з максимальним робочим циклом 30 хвилин на орбіту, а Режим WV працює до максимального робочого циклу 75 хвилин на орбіту.

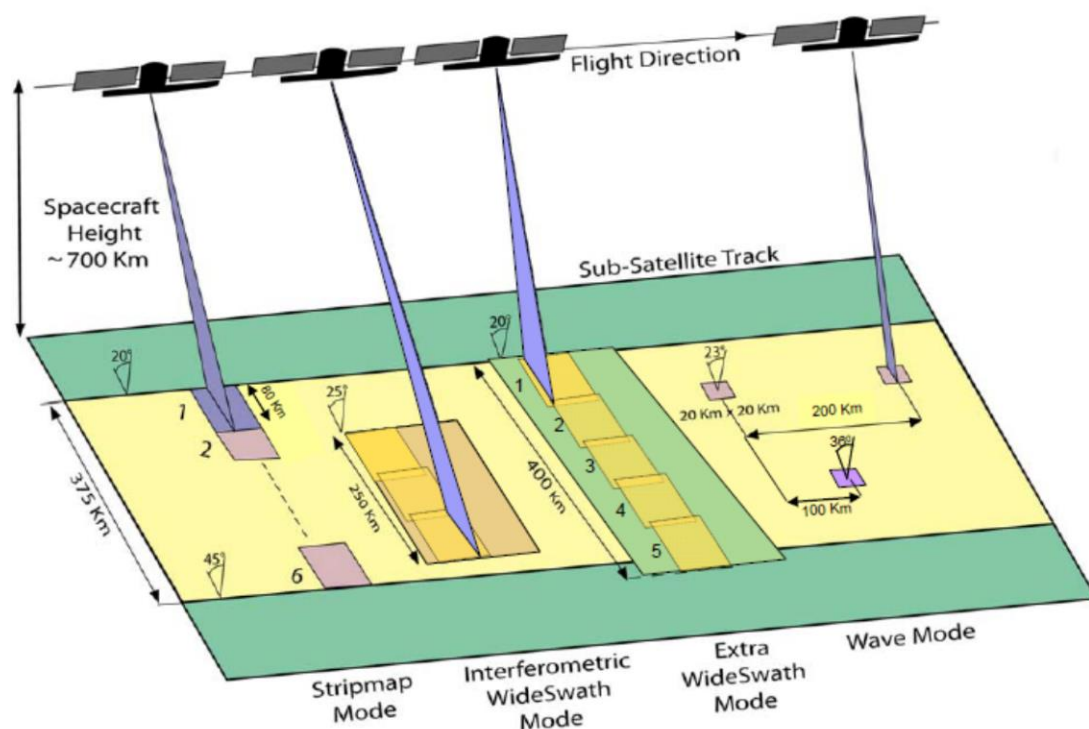


Рис 2.1 Зона покриття різних режимів зйомки Sentinel-1[13]

Інструменти C-band SAR Sentinel-1 підтримують роботу як в одиночній поляризації(HH, VV), так і в подвійній(HH+HV або VV+VH). Над відкритим океаном основний операційний режим зйомки — WV з VV поляризацією, відповідно в майбутньому ми будемо використовувати саме ці дані.

Чутливість Sentinel-1, виражена через шумовий еквівалент сигма-нуль, становить -22 дБ або краще. Радіометрична точність продуктів даних знаходиться в межах статистичної похибки (3σ) 1,0 дБ. Для розподілених цілей коефіцієнт невизначеності становить -22 дБ або краще. Радіометрична точність для Sentinel-1 для всіх режимів вимірювання знаходяться в межах 1 дБ (3σ) і 0,5 дБ (3σ) відповідно. Показник точності відноситься до невизначеності вимірювання коефіцієнта зворотного розсіювання з використанням інтенсивності зображення без урахування ефекту спекл-шуму. Радіометричне розділення стосується невизначеності вимірювань коефіцієнта зворотного розсіювання радара, яка виникає через апаратні шуми та спекл-шум в іншому випадку ідеальної системи. Радіометричне розділення виражається як стандартне відхилення невизначеності вимірювання, нормалізованої стосовно середнього значення. Для системи візуалізації з корекцією зміщення шуму, радіометричне розділення задається формулою 2.1, де σ_{st} - стандартне відхилення інтенсивності зображення в лінійній потужності, I_0 - середня інтенсивність зображення в лінійній потужності, SNR - співвідношення сигнал-шум та L – ефективна кількість відліків.

Ефективна кількість відліків (L) – це параметр, який характеризує "ефективну" кількість незалежних вимірювань, що використовуються для формування одного пікселя в радіолокаційному зображенні. Він визначається як

$L = (\text{середня інтенсивність})^2 / \text{дисперсія інтенсивності}$, відповідно більша кількість незалежних вимірювань (більше L) означає нижчий рівень спекл-шуму та вищу радіометричну роздільну здатність.

$$\frac{\sigma_{st}}{I_0} + \frac{1 + \frac{1}{SNR}}{\sqrt{L}} \quad (2.1)$$

2.2.2 Характеристики супутника Sentinel-2

Місія Sentinel-2 є європейською парою полярноорбітальних супутників (Sentinel-2A і 2B), призначеною для постійного моніторингу земельних і узбережних зон Землі з використанням високороздільних зображень. Система Sentinel-2 базується на одночасній роботі двох ідентичних супутників, які рухаються на одній орбітальній площині, але з фазовим зміщенням на 180° один від одного. Кожен супутник оснащений мультиспектральним оптичним інструментом (MSI), який охоплює спектральний діапазон від видимого до короткохвильового інфрачервоного і забезпечує високороздільні зображення на глобальному рівні з високою частотою повторення. MSI призначений для вимірювання відбитого від Землі випромінювання через атмосферу в 13 спектральних смугах, що охоплюють від видимого і ближнього інфрачервоного (VNIR) до короткохвильового інфрачервоного (SWIR):

- 4 смуги з роздільною здатністю 10 м: синій (490 нм), зелений (560 нм), червоний (665 нм) і ближній інфрачервоний (842 нм).
- 6 смуг з роздільною здатністю 20 м: 4 вузькі смуги для характеристики рослинності (705 нм, 740 нм, 783 нм і 865 нм) і 2 широкі SWIR смуги (1610 нм і 2190 нм) для застосувань, таких як виявлення снігу/льоду/хмар чи оцінка стресу рослинності.
- 3 смуги з роздільною здатністю 60 м, в основному для відсіювання хмар і атмосферних корекцій (443 нм для аерозолів, 945 нм для водяної пари і 1375 нм для виявлення перистих хмар).

Інструмент MSI розроблений з урахуванням вимог великої ширини смуги та високої геометричної і спектральної точності вимірювань. Він базується на телескопі, який живить дві фокусні площини зі спектральною розділеністю. Дві різні групи з 12 оптичних детекторів на кожній фокусній площині покривають VNIR і SWIR канали відповідно.

Режими зйомки:

- 1) Стандартний режим (Standard mode): смуга захоплення 290 км, просторова роздільна здатність від 10 до 60 м в залежності від спектрального каналу.
- 2) Режим точного позиціонування (Narrow Field of View): смуга захоплення 100 км, роздільна здатність від 5 x 20 м.

Висновки по розділу 2

У розділі 2 досліджено сучасні технології спостереження Землі, які використовуються для моніторингу морських екосистем та оцінки стану морського середовища. Використання супутникових систем, оснащених сенсорами з різними довжинами хвиль в електромагнітному спектрі, дозволяє отримувати деталізовані дані про різноманітні аспекти морської поверхні. Активні та пасивні сенсори вимірюють різні властивості, такі як відбиття або поглинання енергії, що дає змогу аналізувати забруднення нафтою, кліматичні зміни та інші екологічні показники.

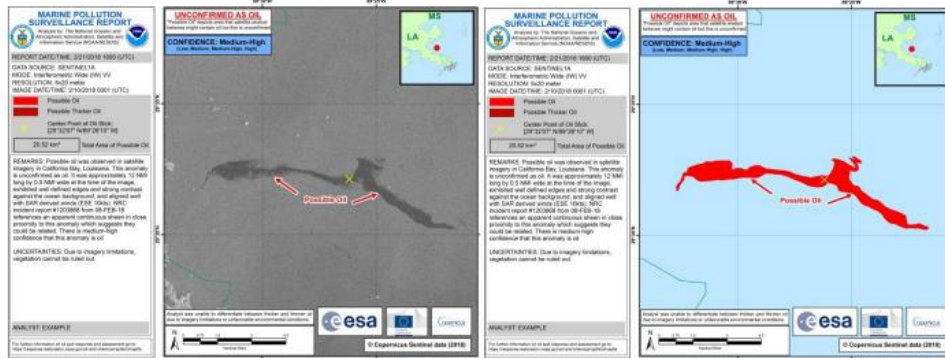
Таблиці 2.1 і 2.2 надають інформацію про різні типи сенсорів і їх характеристики, включаючи обмеження та переваги кожного типу. Пасивні сенсори, які працюють у видимому, інфрачервоному та тепловому інфрачервоному діапазонах, ідеально підходять для денних зйомок у чітких погодних умовах. Водночас активні сенсори, зокрема радар із синтетичною апертурою (SAR), здатні працювати у будь-який час доби та в різних погодних умовах, включаючи хмари та туман. Огляд характеристик супутників Sentinel-1 і Sentinel-2 показує їхні ключові характеристики, режими зйомки та типи продуктів, доступних для моніторингу морського середовища. Sentinel-1 відомий високою роздільною здатністю та здатністю працювати в умовах поганої видимості, що робить його ідеальним для виявлення нафти та моніторингу морських стихій. Sentinel-2, з іншого боку, забезпечує широкий спектр спектральних каналів і підходить для комбінування їх комбінування для дослідження та виведення нових індексів, що сприяє більш глибокому розумінню стану морського середовища.

3 ВИКОРИСТАНІ ДАНІ

Підготовка датасету для моделей семантичної сегментації є ключовим етапом у будь-якому проекті машинного навчання, спрямованому на обробку зображень. Цей розділ детально описує методи та процеси, використані для створення датасету, який включає в себе зображення і відповідні маски сегментації. Головною метою підготовки є забезпечення якісних даних для тренування та оцінки моделей. Для цього використовуються різноманітні джерела, включаючи публічні бази даних супутникових зображень та інші джерела з відкритим доступом.

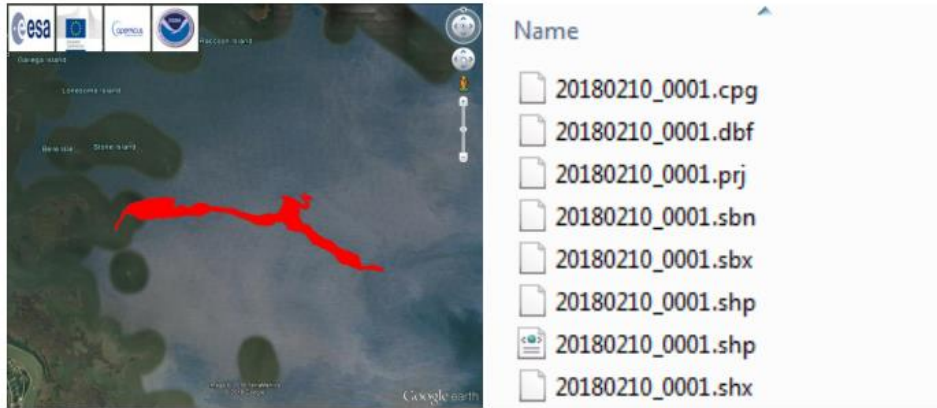
Звіт про спостереження за забрудненням моря (MPSR) — набір даних, створений у відділі супутникового аналізу (SAB) NESDIS, коли у водах США виявлено морську аномалію, яка ймовірно є результатом випадкового або навмисного виливу сирої нафти. Звіти також генеруються для полів морського сміття на основі подій і запитів. Можливі нафтові плями найчастіше виявляють за допомогою аналізу мультиспектральних супутникових зображень і радарів із синтетичною апертурою, але іноді ідентифікують за допомогою інших механізмів спостереження, таких як аерофотозйомка. Ідентифікація аномалій базується на візуальному огляді та різноманітних допоміжних наборах даних, включаючи автоматизований інструмент картографування розливів нафти.

Повний архів MPSR має структуру (рис 3.1) та включає 2 зображення JPEG (рис 3.1), 1 KML, файли GIS і текстовий файл(рис 3.2).



JPEG 1: Satellite image with annotations

JPEG 2: No satellite image with shapefile turned on



KML

GIS Files

Рис. 3.1 – Приклад структури звіту MSPR

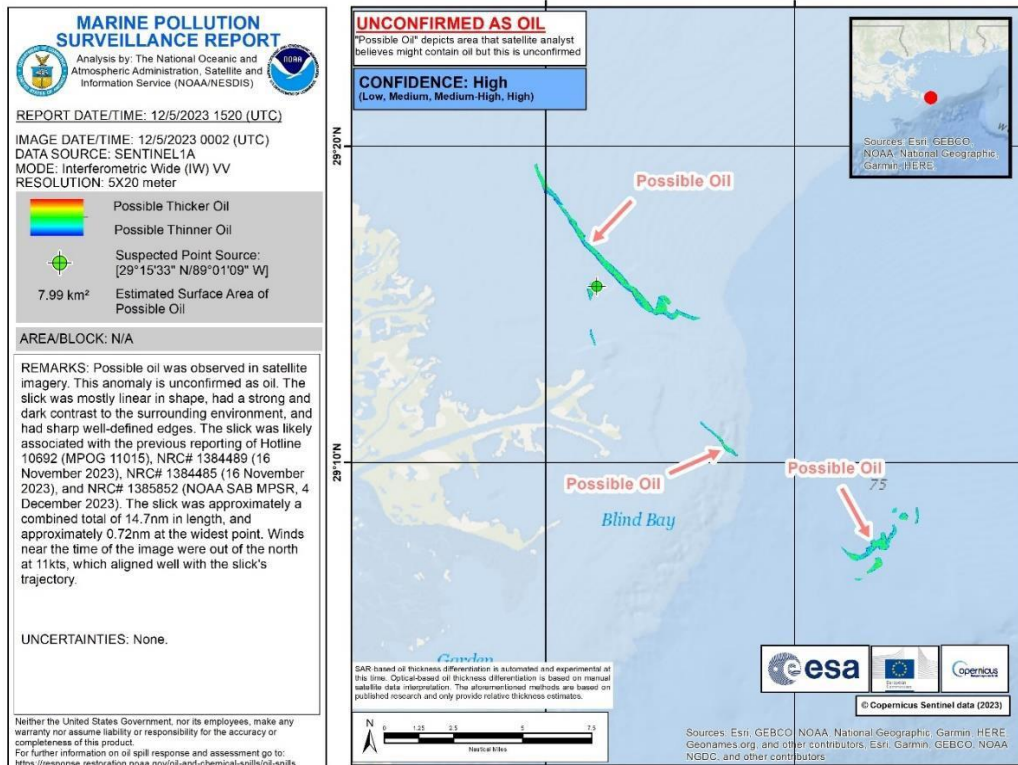


Рис. 3.2 – Приклад зображення звіту MSPR

REPORT DATE: 12-05-2023
REPORT TIME: 1520 UTC
REGION: GULF OF MEXICO
SUB REGION: N/A
AREA/BLOCK: N/A
IMAGE DATE: 12-05-2023
IMAGE TIME: 0002 UTC
DATA SOURCE: SENTINEL1A
MODE: Interferometric Wide (IW) WV
RESOLUTION: 5X20 meter
LOCATION: 29°15'33" N/89°01'09" W
AREA: 7.99 square kilometers
CONFIDENCE: High

Рис 3.3 – Приклад текстового файлу звіту MSPR

Кожен звіт представлений окремим архівом і містить наступну інформацію:

1. Дата та час зображення
2. Датчик супутника, просторова роздільна здатність та режим / поляризація, якщо застосовується
3. Дата та час звіту
4. Оцінка площі поверхні нафтової плями
5. Рівень впевненості на основі внутрішніх критеріїв та практичного досвіду
6. Примітки та невизначеності
7. Похідні картографічні продукти у форматі JPEG
8. Файли ГІС

Відповідно, для створення датасету було завантажено дані за 2011-2024 роки з ресурсу Національного управління океанічних та атмосферних досліджень США (NOAA) [11]. На основі shape-файлів, тобто масок нафтових розливів зі звітів MSPR(рис. 3.4) були створені центроїди наших АОІ для того, щоб не завантажувати весь супутниковий знімок, а лише частину, яка нас цікавить, оскільки об'єм пам'яті фізично не дозволяє завантажити усі супутникові знімки повністю.

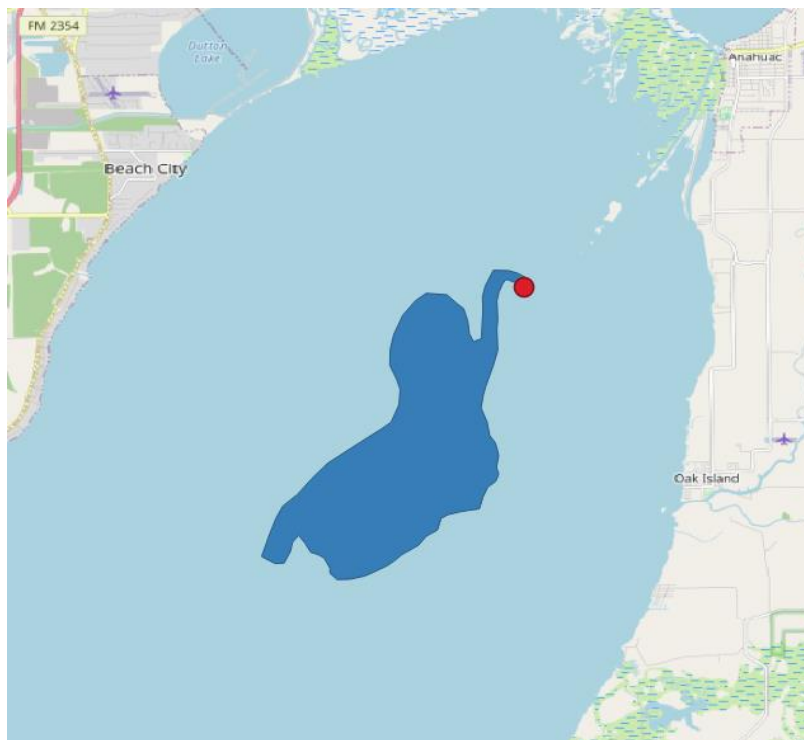


Рис. 3.4 – Shape-файли нафтового розливу

На наступному етапі на основі створеної геометрії, наявних часових даних і даних про впевненість нафтового розливу, було здійснено пошук доступних супутникових продуктів на платформі ASF [12] за допомогою бібліотеки НуРЗ. НуРЗ надає безкоштовні послуги, за допомогою яких користувачі можуть запитувати обробку SAR на вимогу. Ці запити на обробку приймаються автоматизованими системами, які виконують складну обробку SAR від імені користувача. Продукти Sentinel-1 On-Demand RTC від ASF генеруються за допомогою програмного забезпечення GAMMA. Продукти розповсюджуються у вигляді GeoTIFF-файлів (по одному для кожної доступної поляризації), спроектованих на відповідну зону UTM для місцезнаходження нафтового розливу. Відповідно як зазначалось в попередньому розділі, був використаний продукт Level-1 GRD. Перед використанням продуктів SAR для виявлення нафтових розливів за допомогою бібліотеки НуРЗ було проведено певну попередню обробку, тобто радіометричну корекцію, фільтр спекл-шуму, геокодинг.

Було розроблено алгоритм, який включає кілька етапів, які забезпечують ефективно завантаження, розпакування, обробку зображень та створення

необхідних масок для подальшого аналізу. На першому етапі, супутникові знімки завантажуються з визначених URL-адрес або використовуються з локального сховища, якщо вони вже існують. Після завантаження знімки розпаковуються у визначену директорію для подальшої обробки. Наступним кроком є отримання координатної системи супутникового знімка, що забезпечує правильне позиціонування даних на земній поверхні. Далі об'єднуються векторні дані з кількох шейп-файлів, що відповідають одному і тому ж зразку зображення. Ці дані конвертуються до координатної системи супутникового знімка для забезпечення їх сумісності. Після цього, для кожного каналу зображення визначається область інтересу (AOI) на основі векторних даних, що дозволяє обрізати зображення відповідно до заданих параметрів. У наступному етапі створюються обрізані зображення для кожного каналу та зберігаються у визначену директорію. Одночасно з цим, об'єднані векторні дані використовуються для створення растрових масок, які також зберігаються у вигляді файлів GeoTIFF. Для створення міток істини необхідно перетворити векторні маски (shape-файли з MPSR) у растровий формат, що відповідає розміру та геометрії зображення з супутнику, що дозволяє точно визначити області нафтових розливів на рівні пікселів. Для Sentinel-2 були проведені вищевказані етапи, а також було створено зображення на основі індексу OSI(Oil Spill Index).

В результаті було створено датасет з супутникових зображень, який складається з 2471 випадку нафтового розливу, які розділені в пропорції 80%:10%:10%.

Висновки по розділу 3

Розділ 3 надає детальний огляд процесу підготовки датасету для моделей семантичної сегментації на основі супутникових зображень, зокрема Sentinel-1 і Sentinel-2, для виявлення нафтових розливів. Основні джерела даних, такі як звіти про спостереження за забрудненням моря (MPSR) від NOAA та продукти Sentinel-1 від ASF та Sentinel-2 від Sentinel Hub API, використовуються для створення цільового датасету, що включає супутникові зображення і відповідні маски сегментації.

Основні кроки підготовки датасету включають завантаження супутникових зображень з визначених URL або локальних сховищ, їх географічне позиціонування і обрізання з використанням векторних `shape`-файлів `MPSR`. Ці векторні дані конвертуються у растровий формат для створення масок сегментації, які точно визначають області нафтових розливів на рівні пікселів. Для Sentinel-1, продукти обробляються за допомогою `HuP3`, а для Sentinel-2 Sentinel Hub API, що забезпечує необхідну попередню обробку даних для покращення якості зображень перед подальшим аналізом. Підсумовуючи, ми підкреслюємо важливість правильної підготовки даних для моделей машинного навчання, оскільки якість датасету напряму впливає на ефективність моделей семантичної сегментації. Використання різноманітних джерел даних і методів обробки дозволяє створювати достовірні та точні моделі.

4 ПОБУДОВА МОДЕЛЕЙ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ, ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ

Цей розділ присвячений детальному аналізу архітектур та параметрів моделей, що використовуються для сегментації зображень з даних супутників Sentinel 1 та Sentinel 2. У цьому розділі наведено інформацію про функції втрат, які використовувалися для оцінки продуктивності моделей, а також процес вибору оптимальної архітектури та параметрів на основі результатів експериментів. Детально описано використані бібліотеки та інструменти для розробки та навчання моделей, а також аналіз отриманих результатів з використанням найкращих моделей для кожного типу даних.

4.1 Функції втрат

Продуктивність моделей семантичної сегментації оцінювалася на основі двох версій F-міри, які кількісно визначають перекриття між істинним значенням (G) та прогнозом (P). Деталі двох оціночних метрик наведені нижче:

1. Intersection over Union (IoU): Ця метрика вимірює співвідношення площі перетину областей, виділених як модель, до площі їх об'єднання(рис 4.1).

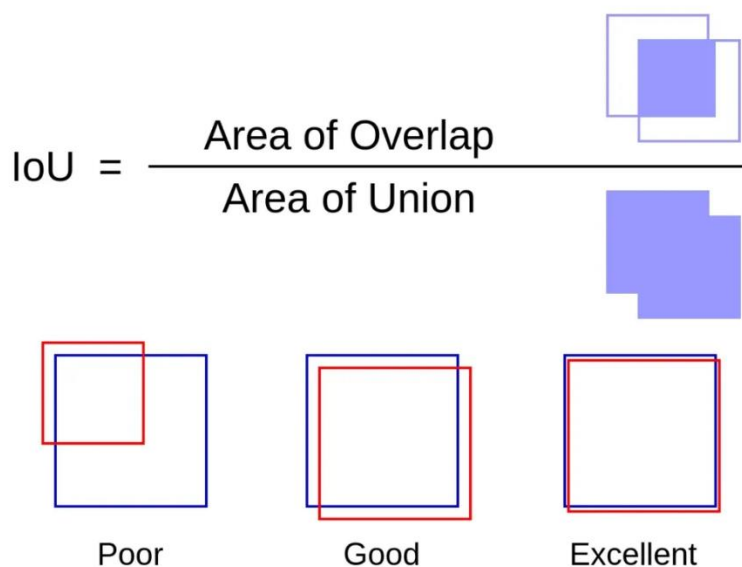


Рис. 4.1 Графічне представлення формули IOU

True Positive (TP): Площа перетину між істинною міткою (Ground Truth, GT) і сегментаційною маскою (S). Математично це логічна операція AND між GT і S

$$TP = GT \cdot S \quad (4.1)$$

False Positive (FP): Прогнозована площа поза істинною міткою. Це логічна операція OR між GT і S мінус GT.

$$FP = (GT + S) - GT \quad (4.2)$$

False Negative (FN): Кількість пікселів в області істинної мітки, яку модель не змогла передбачити. Це логічна операція OR між GT і S мінус S.

$$FN = (GT + S) - S \quad (4.3)$$

Знаючи це, можемо записати IoU як відношення площі перетину до об'єднаної площі передбачення та істинної мітки:

$$IOU = \frac{TP}{(TP+FP+FN)} \quad (4.4)$$

Важливо зазначити, що в задачах сегментації зображень область може мати будь-яку форму, а не обов'язково прямокутну, тому аналіз проводиться піксель за пікселем. Значення IoU понад 0.5 вважається хорошим, оскільки це означає, що більшість пікселів, які були правильно виділені моделлю, збігаються з істинними мітками.

2. F1-Score (Dice Score): F1-Score, також відомий як Dice Score, вимірює точність і реколекцію моделі одночасно. Він обчислюється (формула 4.5) як подвоєне відношення між кількістю правильно визначених пікселів до загальної кількості пікселів, які були виділені моделлю, і загальною кількістю пікселів, які були правильно виділені моделлю та істинними мітками.

$$F1 - Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (4.5)$$

Значення F1-Score понад 0.75 вважається дуже хорошим, оскільки це свідчить про високу точність та реколекцію моделі.

Найкращу модель для даних з кожного супутника будемо обирати на основі моделі з найвищим значенням IoU, а також F1-Score на тестовому наборі. Такий підхід дозволяє вибрати модель, яка не тільки має високу точність у виділенні областей на зображенні, але й має високу здатність до узагальнення на нових даних.

4.2 Вибір моделей та параметрів

Для реалізації та навчання моделей семантичної сегментації було використано PyTorch Lightning, що дозволяє розділити код на модель, оптимізатор та процес навчання і робить код більш зрозумілим. Також, ця бібліотека надає готові рішення для типових завдань таких як валідація, тестування моделі, що відповідно пришвидшує її розробку, підтримує різні архітектури та оптимізатори, і надає ще ряд переваг в масштабуванні і розподіленні обчислень та в моніторингу процесу навчання за допомогою різних callback-функцій і логуювання. В якості експерименту, був обраний цілий ряд найпопулярніших архітектур та їх розширень для сегментації: UNet++, DeepLabV3+ з варіаціями відповідних енкодерів ResNet, EfficientNetB0-8.

Архітектура UNet++ є розширенням класичної архітектури UNet, яка була спеціально розроблена для семантичної сегментації зображень. Основна ідея UNet полягає в тому, щоб мати мережу, яка складається з енкодера (зазвичай з підвіконнями згорткових шарів) та декодера (який використовує транспоновані згорткові шари для поступового збільшення роздільної здатності). Однак оригінальна архітектура UNet може мати обмежену спроможність в розрізненні складних образів через втрату контексту в декодері.

У UNet++, якість сегментації підвищується за рахунок додаткових зв'язків між різними рівнями енкодера та декодера. Замість простого підключення на одному

рівні, UNet++ включає в себе мережі внутрішнього зв'язку, що дозволяють обмінюватися контекстом між різними рівнями (рис 4.2). Це дозволяє зберігати та розширювати просторову інформацію, що поліпшує здатність моделі до точної сегментації образів навіть у випадках з великою кількістю класів або важкими умовами зйомки.

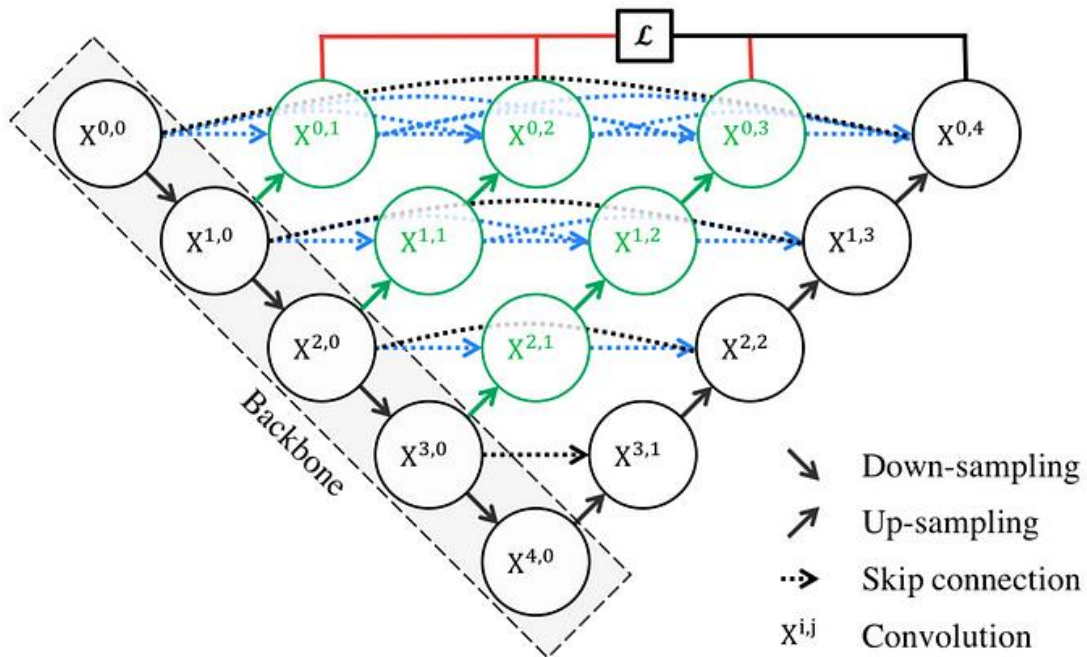


Рис. 4.2 Графічне представлення архітектури UNET++

DeepLabV3+ є ще одним важливим розвитком в галузі семантичної сегментації. DeepLabV3+ є передовою архітектурою для семантичної сегментації зображень, розробленою Google, яка поєднує в собі кілька ключових інноваційних елементів для досягнення високої точності і ефективності.

Основною складовою DeepLabV3+ є Atrous Convolution (ASC)[14].

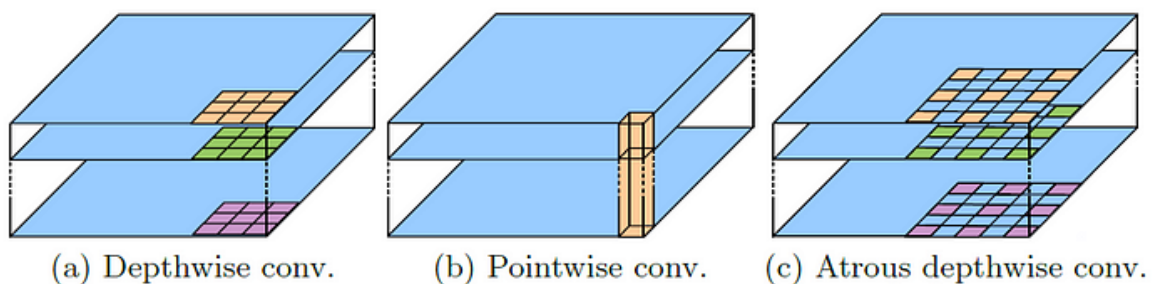


Рис. 4.3 Графічне представлення згорток

DeepLabV3+ з використанням Xception Backbone представляє сучасний підхід до семантичної сегментації зображень. Архітектура зберігає структуру енкодера-

декодера і має кілька ключових характеристик, що забезпечують її ефективність.

Однією з основних характеристик цієї мережі є *atrous convolution* (дилатована згортка), яка працює з вхідною картою ознак (x) за формулою:

$$y(i) = \sum_k x[i + r * k]w[k]$$

де i - це місце у вихідній карті ознак y , w - це згортковий фільтр, а r - це *atrous rate*, який визначає крок вибірки вхідного сигналу. *Atrous convolution* дозволяє збільшити поле зору i , відповідно, контекстуальну інформацію, що обробляється на кожному шарі, з незначним збільшенням кількості параметрів і обчислювальної складності.

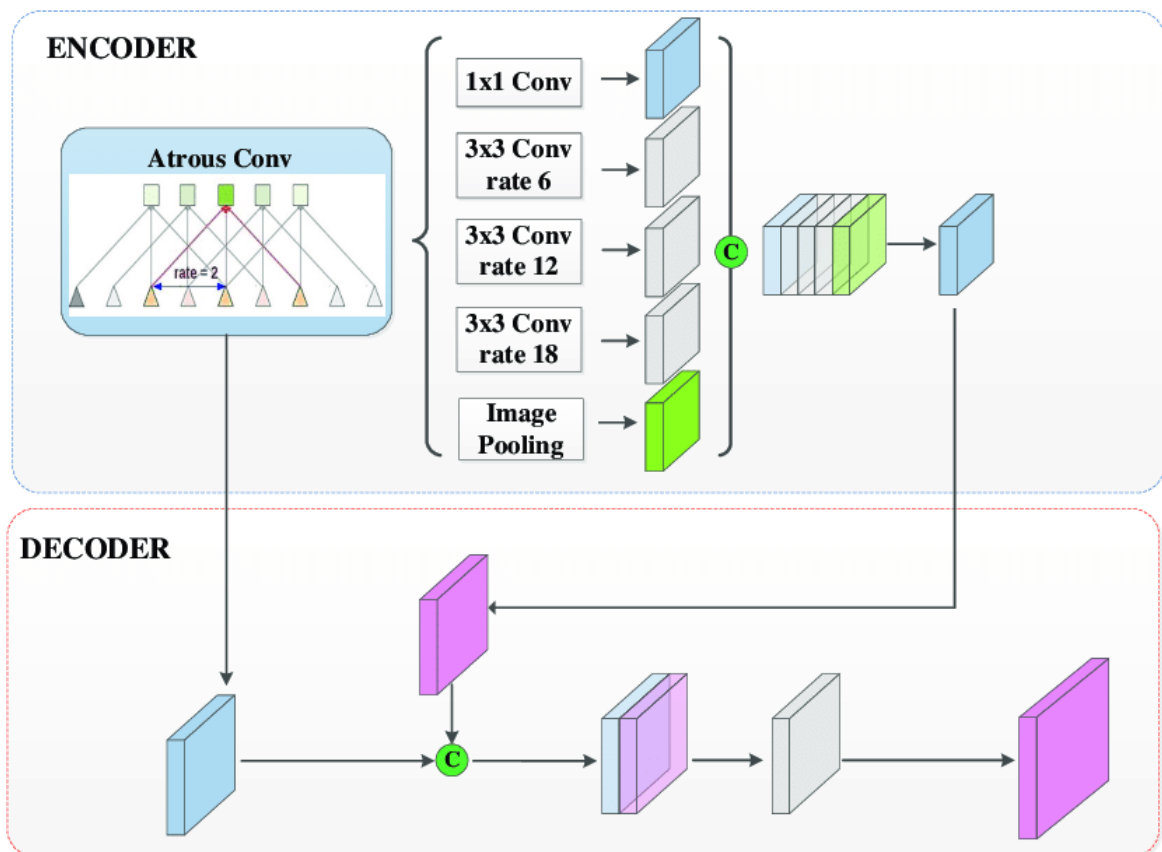


Рис. 4.4 Графічне представлення архітектури DeepLabV3+[15]

Іншою важливою характеристикою є *atrous spatial pyramid pooling (ASPP)*, яка застосовує *atrous convolution* паралельно для вилучення ознак на різних масштабах, що допомагає зменшити втрату просторової інформації через попереднє пулінг або згортки зі стриженням. *ASPP* дозволяє збільшити поле зору i , таким чином, просторовий контекст, що враховується на кожному шарі, без значного збільшення кількості параметрів і обчислювальної складності.

DeepLabv3+ також використовує separable convolutions, введені в попередній версії DeepLabv3[15]. У той час як стандартна згортка виконує обчислення каналів та просторових ознак за один крок, depthwise separable convolution розділяє це обчислення на два кроки: depthwise convolution і pointwise convolution. Depthwise convolution виконує незалежну просторову згортку для кожного вхідного каналу за допомогою лише одного згорткового фільтра, а pointwise convolution використовується для об'єднання виходу depthwise convolutions. Порівняно зі стандартною згорткою, ці операції зменшують кількість параметрів та обчислювальні витрати, зберігаючи подібну точність.

Версія DeepLabv3+ також модифікує модель Xception, замінюючи всі операції max-pooling на depthwise separable convolutions. На етапі декодування ознаки, отримані від енкодера, збільшуються за допомогою білінійного масштабування в 4 рази і потім об'єднуються з відповідними низькорівневими ознаками. Для кращого використання вищих семантичних ознак, витягнутих енкодером, застосовується 1×1 згортка для зменшення кількості каналів. Після об'єднання застосовується 3×3 згортка для покращення ознак, закінчуючи ще одним білінійним масштабуванням в 4 рази для досягнення роздільної здатності вхідного зображення.

Для датасету було використано аугментацію з наступними параметрами для навчального розподілу:

1. Зміна розміру зображення до заданих розмірів 640 x 640.
2. Випадкова зміна яскравості та контрасту (ймовірність: 0.5).
3. Розмивання зображення (ймовірність: 0.5).
4. Випадкове зміщення, масштабування та обертання зображення (ймовірність: 0.5).
5. Горизонтальне відображення (ймовірність: 0.5).
6. Вертикальне відображення (ймовірність: 0.5).
7. Нормалізація за середнім значенням та стандартним відхиленням ImageNet.

Для валідаційного та тестового розподілів:

1. Зміна розміру зображення до заданих розмірів 640 x 640.

2. Нормалізація за середнім значенням та стандартним відхиленням ImageNet.

У всіх моделях було використано оптимізатор Adam з планувальником швидкості навчання ReduceLROnPlateau. В якості додаткових параметрів було також встановлено колбеки:

1. Checkpoint Callback: Зберігає налаштування моделі та параметри/ваги в файл контрольної точки при досягненні найкращих метрик на основі втрат валідації.

2. Learning Rate Monitor Callback: Дозволяє реалізувати планувальник швидкості навчання.

3. Early Stopping Callback: Сигналізує тренеру зупинити процес навчання, якщо не було покращення найкращих втрат валідації протягом 10 послідовних епох.

А також, використано логгер TensorboardLogger для збереження метрик через певні інтервали і було протестовано автомасштабування розміру батчу і половинну точність або комбіновану точність, що дозволяє зменшити обсяг пам'яті під час навчання моделі і подолати обмеження пам'яті GPU.

Було реалізовано алгоритм пошуку швидкості навчання (Learning Rate Finder), який автоматично визначає оптимальну початкову швидкість для навчання.

4.3 Аналіз результатів роботи моделей

Для аналізу результатів роботи моделей на основі тестового датасету було обрано по одній моделі для кожного супутника, які показали найкращі метрики.

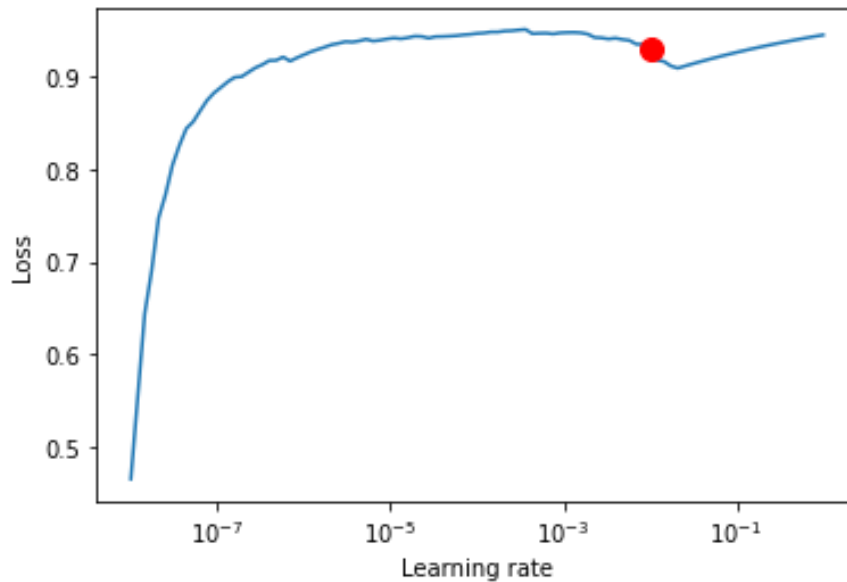


Рис. 4.5 Візуалізація алгоритму пошуку швидкості навчання

Як видно на графіку(рис 4.5), оптимальна швидкість навчання для моделі UNet++ 0.01, хоча початково модель була ініціалізована з швидкістю навчання 0.001.

Для Sentinel 1 найефективнішою моделлю виявилась модель з наступними характеристиками:

- Архітектура моделі: UNet++
- Енкодер: EfficientNetB3
- Розмір батча (Batch Size): 4
- Initial Learning Rate: 0.001
- Кількість епох: 100
- Optimizer: Adam
- Loss Function: Dice Loss

І показала наступні результати:

'test_dataset_f1': 0.77,
 'test_dataset_iou': 0.64,
 'test_loss': 0.26,
 'test_per_image_f1': 0.67,
 'test_per_image_iou': 0.51

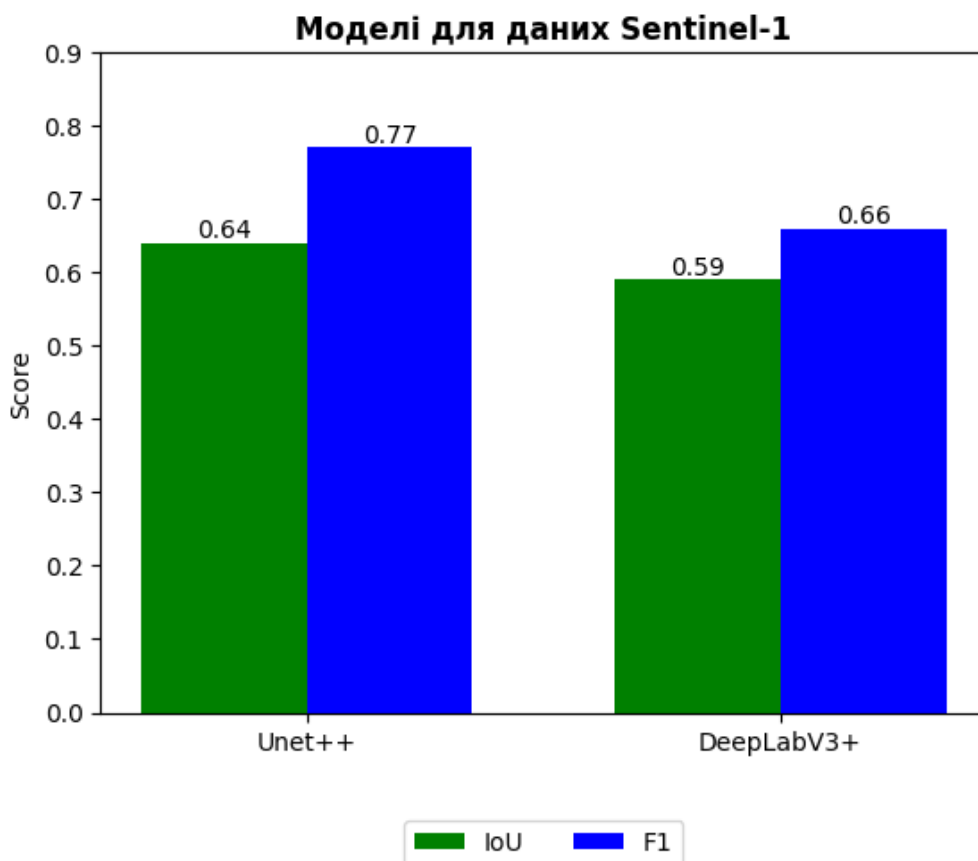


Рис 4.6 – Порівняння архітектур моделей для даних Sentinel-1

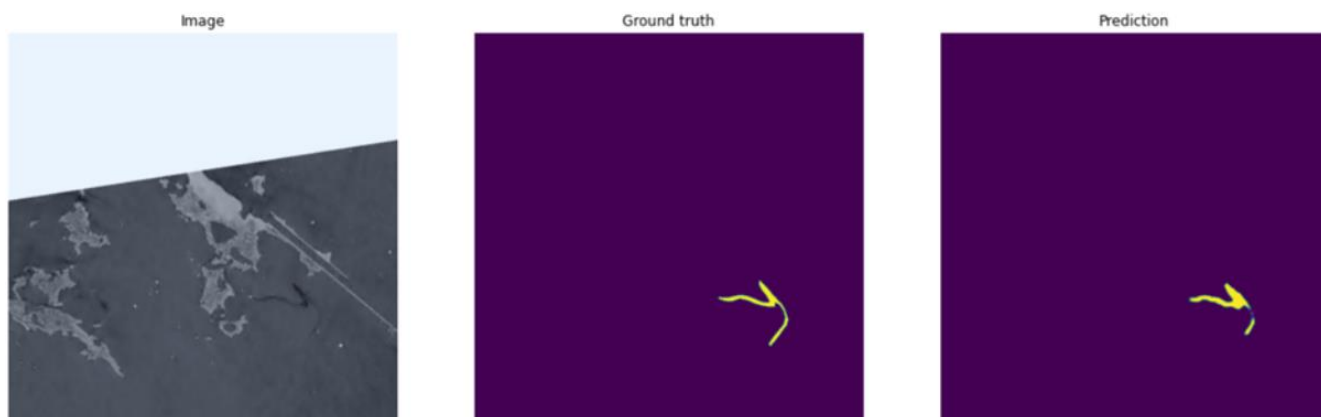


Рис 4.7 – Результат роботи моделі на архітектурі UNet++

Модель на основі UNet++ показала високі метрики якості сегментації IoU та F1 Score. Використання архітектури UNet++ та енкодера EfficientNetB3 дозволило отримати точні та надійні результати, а розмір батчу 4 дозволяє ефективно використовувати ресурси пам'яті та обчислювальні потужності.

Для Sentinel 2 найефективнішою моделлю виявилась модель з наступними характеристиками:

- IoU Score: 0.45
- F1 Score (Dice Score): 0.61
- Архітектура моделі: DeepLabV3+
- Енкодер: EfficientNetB0
- Розмір пакету (Batch Size): 16
- Initial Learning Rate: 0.001
- Кількість епох: 100
- Optimizer: Adam
- Loss Function: Dice Loss

І показала наступні результати:

test_dataset_f1': 0.61,
'test_dataset_iou': 0.45,
'test_loss': 0.33,
'test_per_image_f1': 0.59,
'test_per_image_iou': 0.51

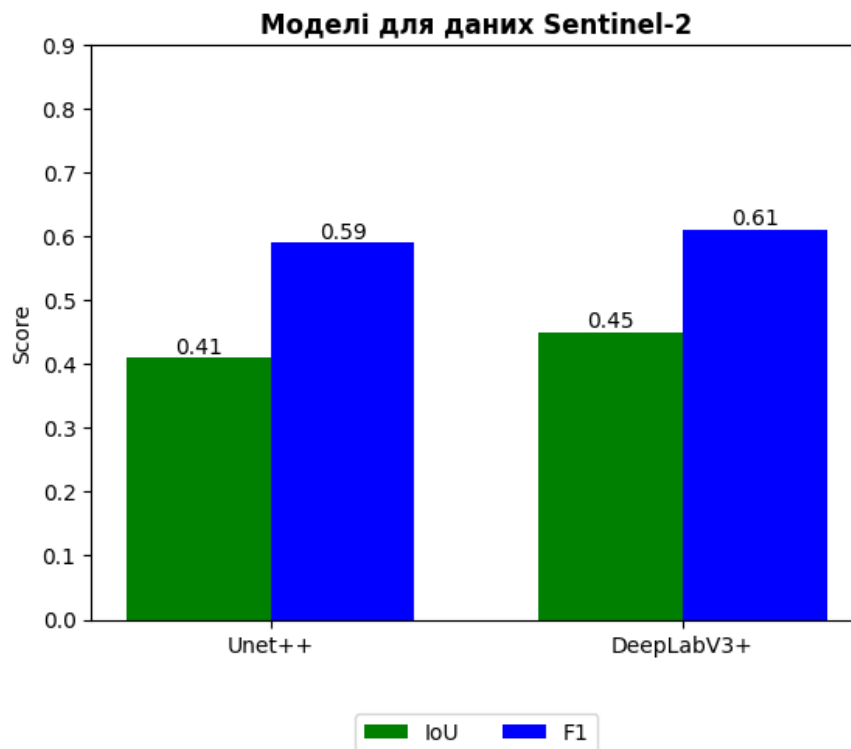


Рис 4.8 – Порівняння архітектур моделей для даних Sentinel-1

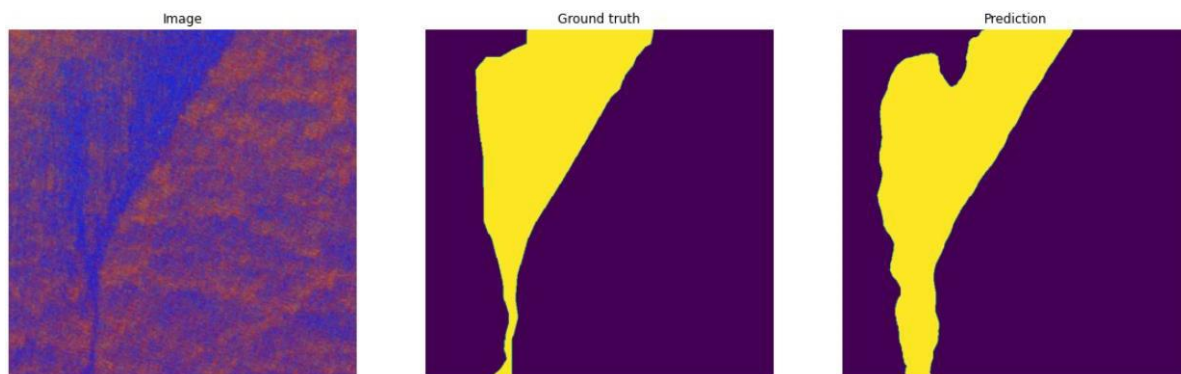


Рис 4.9 – Результат роботи моделі на архітектурі DeepLabV3+ у порівнянні з моделлю для даних Sentinel-1, ця модель показала гірші метрики якості сегментації. Хоча використання архітектури DeepLabV3+ та енкодера EfficientNetB0 дозволило отримати деяку точність, але результати все ще залишаються нижчими порівняно з першою моделлю. Розмір пакету 16 може бути великим для ефективного використання ресурсів, що може призвести до погіршення якості навчання та сегментації.

Висновки по розділу 4

У цьому розділі проведено інтенсивний аналіз різних архітектур та параметрів моделей для семантичної сегментації зображень, отриманих з супутників Sentinel 1 та Sentinel 2. Використання двох ключових метрик оцінки — Intersection over Union (IoU) і F1-Score (Dice Score) — дозволило об'єктивно порівняти продуктивність моделей. Підбір оптимальних параметрів, таких як енкодер (EfficientNetB3 для Sentinel 1 і EfficientNetB0 для Sentinel 2), розмір батчу і кількість епох навчання, був ключовим для досягнення високих показників точності. Модель на основі архітектури UNet++ з енкодером EfficientNetB3 продемонструвала найкращі результати для даних Sentinel 1, досягаючи $\text{IoU} = 0.64$ і $\text{F1-Score} = 0.77$. В той же час, модель з архітектурою DeepLabV3+ і енкодером EfficientNetB0 для Sentinel 2 показала менші, але все ще значимі показники, з $\text{IoU} = 0.45$ і $\text{F1-Score} = 0.61$. Висновки підкреслюють важливість вибору належної архітектури з урахуванням особливостей вихідних даних для оптимізації точності семантичної сегментації. Результати експериментів підтверджують ефективність використання сучасних модифікацій UNet і DeepLabV3+ з відповідними енкодерами в контексті обробки зображень з супутників.

ВИСНОВКИ

У даному дослідженні проведено аналіз джерел та методів семантичної сегментації зображень для виявлення нафтових плям у водах, заснований на даних звітів про спостереження за забрудненням моря (MPSR) та супутникових знімках. Процес дослідження і експериментів включали в себе наступні етапи:

1) Огляд опублікованих джерел та документів за тематикою дослідження, зокрема робіт, присвячених використанню супутникових даних для виявлення нафтових забруднень у водах, методів семантичної сегментації та їх застосування у сфері охорони довкілля.

2) Підготовлено датасет, що складається зі звітів MPSR та супутникових зображень, та імплементація обраних моделей для сегментації нафтових плям. Використано бібліотеку PyTorch Lightning для реалізації та навчання моделі.

3) Було проаналізовано популярні архітектури для семантичної сегментації, такі як UNet++ та DeepLabV3+, з різними енкодерами, включаючи EfficientNetB0-8. Ці архітектури забезпечують різні підходи до обробки контекстуальної інформації та збереження просторових ознак, а також функцій втрат для семантичної сегментації. В якості метрик оцінки продуктивності моделей використано Intersection over Union (IoU) та F1-Score (Dice Score).

4) Проведено аналіз отриманих результатів, включаючи порівняння різних архітектур моделей та їх продуктивності за метриками IoU та F1-Score. Також вивчено вплив параметрів навчання, таких як швидкість навчання та розмір пакету, на результати моделі.

Результати дослідження дозволили визначити оптимальні моделі для різних супутників та параметри навчання для семантичної сегментації нафтових плям у водах на основі супутникових зображень, що може бути використано для моніторингу забруднень морського середовища та прийняття заходів щодо їх ліквідації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. United Nations Environment, Mediterranean Action Plan “*2017 Mediterranean Quality Status Report*”, англ (), url: <https://www.unep.org/unepmap/resources/quality-status-report-mediterranean-med-qsr-2017>
2. J. Ma, R. Ma, Qi Pan, X. Liang, J. Wang, X. Ni “A Global Review of Progress in Remote Sensing and Monitoring of Marine Pollution”, англ, в *Water*, жовт. 2023, DOI: 10.3390/w15193491, url: <https://doi.org/10.3390/w15193491>
3. Q. Zhong, X.M. Liu “*Monitoring Methods of Marine Pollution Range Based on Big Data Technology*”, англ, в *Nature Environment and Pollution Technology* 20, груд. 2021, DOI: 10.46488/NEPT.2021.v20i04.050, url: <http://dx.doi.org/10.46488/NEPT.2021.v20i04.050>
4. ІПІЕСА-ІОГР “*Satellite remote sensing of oil spills at sea*”, англ (), 2016, url: <https://www.ipieca.org/resources/satellite-remote-sensing-of-oil-spills-at-sea>
5. P. Tysi c, T. Strelets, W. Tuszy nska “*The Application of Satellite Image Analysis in Oil Spill Detection*” англ., в *Applied Sciences*, квіт. 2022, DOI: 10.3390/app12084016, url: <https://doi.org/10.3390/app12084016>
6. J. Yang, Yi Ma, Y. Hu, Z. Jiang, J. Zhang, J. Wan, Z. Li “*Decision Fusion of Deep Learning and Shallow Learning for Marine Oil Spill Detection*” англ., в *Remote Sens.*, січ. 2022, DOI: 10.3390/rs14030666, url: <https://doi.org/10.3390/rs14030666>
7. J. Shi, J. Jiao, Y. Lu, M.Zhang, Z. Mao, Y. Liu “*Determining spectral groups to distinguish oil emulsions from Sargassum over the Gulf of Mexico using an airborne imaging spectrometer*” англ., в *ISPRS Journal of Photogrammetry and Remote Sensing*, груд. 2019, DOI:10.1016/j.isprsjprs.2018.09.017, url: <https://doi.org/10.1016/j.isprsjprs.2018.09.017>
8. M. Krestenitis ,G. Orfanidis,K. Ioannidis,K. Avgerinakis,S. Vrochidis, I.Kompatsiaris “*Oil Spill Identification from Satellite Images Using Deep Neural Networks*” англ., в *Remote Sens.*, лип. 2019, DOI: 10.3390/rs11151762, url: <https://doi.org/10.3390/rs11151762>

9. Xinzhe Wang, Jiaxu Liu, Shuai Zhang, Qiwen Deng, Zhuo Wang, Yunhao Li, Jianchao Fan “ *Detection of Oil Spill Using SAR Imagery Based on AlexNet Model*” *англ.*, в Computational Intelligence and Neuroscience, лип. 2021, DOI: 10.1155/2021/4812979, url: <https://doi.org/10.1155/2021/4812979>
10. S. Rajendran, P. Vethamony, F. Sadooni, H. Al-Saad Al- Kuwari “*Sentinel-2 image transformation methods for mapping oil spill – A case study with Wakashio oil spill in the Indian Ocean, off Mauritius*”, *англ.*, в MethodsX, бер. 2021, DOI:10.1016/j.mex.2021.101327, url: <http://dx.doi.org/10.1016/j.mex.2021.101327>
11. “NOAA” *англ.* (), url: <https://www.ospo.noaa.gov>
12. “Alaska satellite facility.” *англ.* (), url: <https://asf.alaska.edu/>.
13. “Sentinel Wiki” *англ.* (), url: <https://sentiwiki.copernicus.eu>
14. Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam, “*Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*”, *англ.*, 2018, DOI: 10.48550/arXiv.1802.02611, url:<https://doi.org/10.48550/arXiv.1802.02611>
15. Daliana Lobo Torres, Raul Feitosa, Patrick Nigri Happ, Laura Elena Cue La Rosa “*Applying Fully Convolutional Architectures for Semantic Segmentation of a Single Tree Species in Urban Environment on High Resolution UAV Optical Imagery*”, *англ.*, в Sensors 20, DOI: 10.3390/s20020563, url: <https://www.mdpi.com/1424-8220/20/2/563>
16. L. Shumilo, A. Okhrimenko, N. Kussul, S. Drozd, and O. Shkalikov, “*Generative adversarial network augmentation for solving the training data imbalance problem in crop classification*,” Remote Sensing Letters, vol. 14, no. 11, pp. 1129-1138, 2023.
17. N. N. Kussul et al., “*Crop Yield Forecasting Regression Models based on MODIS Data*” Current Problems in RemoteSensing of the Earth from Space, vol. 9, no. 1, pp. 95-107, 2
18. Lavreniuk, M., Shumilo, L., Yailymov, B., & Kussul, N. (2022). *Reviewing Deep Learning Methods in the Applied Problems of Economic Monitoring Based on Geospatial Data*. Cybernetics and Systems Analysis, 58(6), 1008-1020.
19. Kussul Nataliia, Kuzin Volodymyr, Salii Yevhenii, Yailymov Bohdan, Shelestov Andrii *Transfer Learning for Oil Spills Detection Based on Satellite Data*. 4th International

Symposium on Applied Geoinformatics (ISAG2024), May 9 - 10, 2024 (accepted)

20. Henitsoi Pavlo, Shelestov Andrii *Transfer Learning Model for Chlorophyll-a Estimation Using Satellite Imagery*. 4th International Symposium on Applied Geoinformatics (ISAG2024), May 9 - 10, 2024 (accepted)

21. Nataliia Kussul, Volodymyr Kuzin, Yevhenii Salii, Bohdan Yailymov, Andrii Shelestov *Transfer learning approach for oil spills' detection using SAR satellite data*. 2024 IEEE 42nd International Conference on Electronics and Nanotechnology (ELNANO), May 13 – 16, 2024 (accepted)

22. Bohdan Yailymov, Pavlo Henitsoi, Nataliia Kussul, Andrii Shelestov *Increasing Chlorophyll-A Spatial Resolution Using Machine Learning*. IEEE International Conference on System Analysis & Intelligent Computing (SAIC), October 08 – 11, 2024 (submitted)

23. Bohdan Yailymov, Nataliia Kussul, Pavlo Henitsoi, Andrii Shelestov *Improving the spatial resolution of chlorophyll-a in the Mediterranean Sea based on machine learning*. Journal Radioelectronic and Computer Systems (submitted)

ДОДАТОК А ТЕКСТИ ПРОГРАМ

A1 Завантаження даних Sentinel-1

```
import pandas as pd
import hupu as hup

def build_sentinel_list(input_path, output_path):
    data_df1 = pd.read_csv(input_path, header='infer', parse_dates=True)

    sentinel_df = pd.DataFrame(columns=["imagery_id", "satellite",
                                       "capture_date",
                                       "product_count", "product_name", "is_sentinel",
                                       "job_name", "job_status", "download_status",
                                       "file_name", "file_size", "file_url", "shp_files",
                                       "group_id", "group_shp_files", "bands", "mask",
                                       "labels",
                                       "image_status"])

    sentinel_df["imagery_id"] = data_df1[["imagery_id"]]
    sentinel_df["satellite"] = data_df1[["satellite"]]
    sentinel_df["capture_date"] = data_df1[["capture_date"]]
    sentinel_df["product_count"] = data_df1[["product_count"]]
    sentinel_df["product_name"] = data_df1[["product_name"]]
    sentinel_df["shp_files"] = data_df1[["shp_files"]]
    sentinel_df["group_id"] = data_df1[["group_id"]]
    sentinel_df["group_shp_files"] = data_df1[["group_shp_files"]]

    sentinel_df["is_sentinel"] = ["SENTINEL1" in str(i).upper() for i in
                                  sentinel_df.satellite.tolist()]

    sentinel_df.drop(sentinel_df[~(sentinel_df.is_sentinel)].index,
                     inplace=True)
```

```

sentinel_df.to_csv(output_path, index=True, index_label="index")

def submit_jobs(output_path, prefix, mode='NORMAL'):
    sentinel_df = pd.read_csv(output_path, header="infer", parse_dates=True)

    prefix_error = False

    for index, row in sentinel_df[(sentinel_df.product_count > 0) &
    (sentinel_df.job_name.isna())].iterrows():
        job_name = f"{prefix}-{row.index}"
        job_details = hyp.find_jobs(name=job_name)
        if len(job_details) > 0 and mode == 'NORMAL':
            print(f"Prefix error or input file may not be the latest.")
            print(f"Job previously submitted: {job_name}, index: {row.index}")
            print(f"Product Name: {row.product_name.split(',')[0]}")
            print(f"Job Granule: {job_details[0].job_parameters['granules'][0]}")
            prefix_error = True
            break
        elif len(job_details) > 0 and mode == 'INCREMENTAL':
            sentinel_df.loc[index, "job_name"] = job_name
            sentinel_df.loc[index, "job_status"] = job_details[0].status_code

    if mode == 'INCREMENTAL':
        print("Updating previously submitted job details on the same index")
        sentinel_df.to_csv(output_path, index=False)

    if prefix_error:
        print("Exiting process without submitting jobs; resolve errors and re-
        submit if necessary")
        return None
    else:

```

```

print("Valid prefix. Submitting jobs now ...")

success_counter = 0
error_counter = 0

for index, row in sentinel_df[(sentinel_df.product_count > 0) &
(sentinel_df.job_name.isna())].iterrows():
    job_name = f"{prefix}-{row.index}"
    try:
        job = hyp.submit_job(name=job_name,
                             granule=row.product_name.split(',')[0],
                             radiometry='sigma0',
                             scale='amplitude',
                             speckle_filter=True)

        job_status = "SUBMITTED"
        success_counter += 1

        print(f"Submitted: {job_name} for {row.imagery_id}:
{row.product_name.split(',')[0]}")
    except:
        job_status = "NOT SUBMITTED"
        error_counter += 1

        print(f"Error: {job_name} for {row.imagery_id}:
{row.product_name.split(',')[0]}")
        job_name = ""
    finally:
        sentinel_df.loc[index, "job_name"] = job_name
        sentinel_df.loc[index, "job_status"] = job_status

sentinel_df.to_csv(output_path, index=False)

print(f"Successful jobs: {success_counter}")
print(f"Failed jobs: {error_counter}")

```

```

def check_jobs_status(output_path):
    sentinel_df = pd.read_csv(output_path, header="infer", parse_dates=True)

    for index, row in sentinel_df[((sentinel_df.product_count > 0) &
                                   (~sentinel_df.job_name.isna()) &
                                   (~sentinel_df.job_status.isin(['ERROR',
                                                                    'SUCCEEDED'])))].iterrows():
        job_details = hyp.find_jobs(name=row.job_name)

        sentinel_df.loc[index, "job_status"] = job_details[0].status_code

        if job_details[0].succeeded():
            sentinel_df.loc[index, "file_name"] =
            job_details[0].files[0]["filename"]
            sentinel_df.loc[index, "file_size"] = job_details[0].files[0]["size"]
            sentinel_df.loc[index, "file_url"] = job_details[0].files[0]["url"]

    sentinel_df.to_csv(output_path, index=False)

    print(f"Status Summary:")
    print(f"{sentinel_df.groupby(by=['job_status']).job_status.count()}")

    build_sentinel_list("input_manifest.csv", "sentinel_list.csv")
    submit_jobs("sentinel_list.csv", "job_prefix", mode='NORMAL')
    check_jobs_status("sentinel_list.csv")

```

A2 Завантаження даних Sentinel-2

```

import pandas as pd
import ast
import os
from datetime import datetime
from sentinelhub import BBox, CRS, bbox_to_dimensions, SentinelHubRequest,
DataCollection, MosaickingOrder, MimeType

def load_csv(csv_path):
    data = pd.read_csv(csv_path)
    return data

def process_shape_coordinates(shape_coordinates):
    shape_coord_lists = []
    for shape in shape_coordinates:
        if shape == '[]':
            shape_coord_lists.append([])
        else:
            coord = ast.literal_eval(shape)
            if len(coord) != 0:
                shape_coord_lists.append(coord[0])
    return shape_coord_lists

def filter_sentinel_ids(data):
    allowed_satellites = ["SENTINEL2A", "SENTINEL2B", "AQUA,SENTINEL2A",
"SENTINEL2BT16RBC", "LANDSAT9,SENTINEL2B"]
    sentinel_ids = list(data[data['satellite'].isin(allowed_satellites)][['index']])
    return sentinel_ids

def convert_bytes(size):
    for x in ['bytes', 'KB', 'MB', 'GB', 'TB']:
        if size < 1024.0:
            return "%3.1f %s" % (size, x)
        size /= 1024.0

def create_extended_bbox(coord_list, k):
    if coord_list[0] >= 0 and coord_list[2] >= 0:
        x_diff = coord_list[2] - coord_list[0]
    elif coord_list[0] < 0 and coord_list[2] < 0:
        x_diff = coord_list[0] - coord_list[2]
    elif (coord_list[0] < 0 and coord_list[2] > 0) or (coord_list[0] > 0 and coord_list[2] <
0):
        raise TypeError('xmin and xmax contain different signs due to shape being around
0-axis. Check coordinates.')

```

```

coord_list[0] = coord_list[0] - k * abs(x_diff)
coord_list[2] = coord_list[2] + k * abs(x_diff)

if coord_list[3] >= 0 and coord_list[1] >= 0:
    y_diff = coord_list[3] - coord_list[1]
elif coord_list[3] < 0 and coord_list[1] < 0:
    y_diff = coord_list[1] - coord_list[3]
elif (coord_list[0] < 0 and coord_list[2] > 0) or (coord_list[0] > 0 and coord_list[2] <
0):
    raise TypeError('ymin and ymax contain different signs due to shape being around
0-axis. Check coordinates.')

coord_list[1] = coord_list[1] - k * abs(x_diff)
coord_list[3] = coord_list[3] + k * abs(x_diff)

return coord_list

def save_tiff (data, index, resolution):
    file_name = data[data['index'] == index]['imagery_sample_id'].iloc[0]
    dest_folder = 'Images/SENTINEL2/%s' % file_name

    if os.path.isdir('Images/SENTINEL2/%s' % file_name):
        return "Product already downloaded!"

    shape_coordinates = process_shape_coordinates(data ['shape_extent'])
    if len(shape_coordinates[index-1]) == 0:
        return "Instance does not have a shape extent"

    capture_datetime = data [data ['index'] == index]['capture_datetime'].iloc[0]
    capture_datetime = datetime.strptime(capture_datetime, '%d-%m-%Y %H:%M')
    capture_date = capture_datetime.strftime("%Y-%m-%d")

    capture_bbox = BBox(bbox=shape_coordinates[index-1], crs=CRS.WGS84)
    capture_size = bbox_to_dimensions(capture_bbox, resolution=resolution)

    evalscript_all_bands = """
//VERSION=3
function setup() {
    return {
        input: [{
            bands:
["B01", "B02", "B03", "B04", "B05", "B06", "B07", "B08", "B8A", "B09", "B10", "B11", "B12
"],
            units: "DN"

```

```

    }],
    output: {
        bands: 13,
        sampleType: "INT16"
    }
};
}

```

```

function evaluatePixel(sample) {
    return [sample.B01,
            sample.B02,
            sample.B03,
            sample.B04,
            sample.B05,
            sample.B06,
            sample.B07,
            sample.B08,
            sample.B8A,
            sample.B09,
            sample.B10,
            sample.B11,
            sample.B12];
}
"""

```

```

request_all_bands = SentinelHubRequest(
    data_folder=dest_folder,
    evalscript=evalscript_all_bands,

```

```

input_data=[SentinelHubRequest.input_data(data_collection=DataCollection.SENTINE
L2_L1C,
                                                time_interval=(capture_date, capture_date),
                                                mosaicking_order=MosaickingOrder.LEAST_CC,
                                                )
            ],
    responses=[SentinelHubRequest.output_response("default", MimeType.TIFF)],
    bbox=capture_bbox,
    size=capture_size,
    config=config,
)

```

```

request_all_bands.save_data()

```

```

tiff_path = ""
for dirpath, _, files in os.walk(dest_folder, topdown=False):

```

```

for file in files:
    if os.path.abspath(os.path.join(dirpath, file))[-5:] == '.tiff':
        tiff_path = os.path.join(dirpath, file)
if tiff_path == "":
    raise TypeError("No .tiff file detected")

with rasterio.open(tiff_path) as src:
    out_images = src.read()
    out_meta = src.meta

size = os.path.getsize(tiff_path)

return out_meta, out_images

data1 = load_csv('123.csv')
index = 1
resolution = 10
save_tiff(data1, index, resolution)

A3 Створення bbox
import numpy as np
import geopandas as gpd
from shapely.geometry import Polygon
import matplotlib.pyplot as plt

def get_simple_crop_bbox(band, shape, xpixels, ypixels, stretch_factor, plot_box,
rseed=0):
    try:
        if rseed > 0:
            np.random.seed(seed=rseed)

        xmin, ymin, xmax, ymax = band.rio.bounds()
        xres, yres = np.abs(band.rio.resolution())
        sat_crs = band.rio.crs

        if plot_box:
            sat_bbox = [Polygon([[xmin, ymin], [xmax, ymin], [xmax, ymax], [xmin,
ymax]])]
            sat_bbox_gdf = gpd.GeoDataFrame(sat_bbox, columns=["geometry"],
crs=sat_crs)

            oxmin, oymin, oxmax, oymax = shape.total_bounds.tolist()

            oxmin = xmin if oxmin < xmin else oxmin

```

```

oxmax = xmax if oxmax > xmax else oxmax
oymin = ymin if oymin < ymin else oymin
oymax = ymax if oymax > ymax else oymax

if plot_box:
    oil_bbox = [Polygon([[oxmin, oymin], [oxmax, oymin], [oxmax, oymax],
[oxmin, oymax]])]
    oil_bbox_gdf = gpd.GeoDataFrame(oil_bbox, columns=["geometry"],
crs=sat_crs)

    oxcent, oycent = [oxmin + ((oxmax - oxmin) / 2), oymin + ((oymax - oymin) / 2)]

    if ((oxmax - oxmin) / xres) > xpixels or ((oymax - oymin) / yres) > ypixels:
        print("\t\tWarning: Oil spill bounding box is larger than required image
dimensions; so it will be cropped")

    pb_xmin_oxcent = int(np.floor((oxcent - xmin) / xres))
    pb_xmax_oxcent = int(np.floor((xmax - oxcent) / xres))
    pb_ymin_oycent = int(np.floor((oycent - ymin) / yres))
    pb_ymax_oycent = int(np.floor((ymax - oycent) / yres))

    xstretch = int(np.floor(xpixels * stretch_factor))
    ystretch = int(np.floor(ypixels * stretch_factor))
    xstretchleft = xstretchright = xstretch
    ystretchbottom = ystretchtop = ystretch

    if pb_xmin_oxcent < xstretch and pb_xmax_oxcent > xstretch:
        xstretchleft = pb_xmin_oxcent
        xstretchdiff = xstretch - xstretchleft
        xstretchright = (xstretch + xstretchdiff) if pb_xmax_oxcent > (xstretch +
xstretchdiff) else pb_xmax_oxcent
    elif pb_xmin_oxcent < xstretch and pb_xmax_oxcent <= xstretch:
        xstretchleft = pb_xmin_oxcent

    if pb_xmax_oxcent < xstretch and pb_xmin_oxcent > xstretch:
        xstretchright = pb_xmax_oxcent
        xstretchdiff = xstretch - xstretchright
        xstretchleft = (xstretch + xstretchdiff) if pb_xmin_oxcent > (xstretch +
xstretchdiff) else pb_xmin_oxcent
    elif pb_xmax_oxcent < xstretch and pb_xmin_oxcent <= xstretch:
        xstretchright = pb_xmax_oxcent

    if pb_ymin_oycent < ystretch and pb_ymax_oycent > ystretch:
        ystretchbottom = pb_ymin_oycent
        ystretchdiff = ystretch - ystretchbottom

```

```

    ystretchtop = (ystretch + ystretchdiff) if pb_ymax_oycent > (ystretch +
ystretchdiff) else pb_ymax_oycent
    elif pb_ymin_oycent < ystretch and pb_ymax_oycent <= ystretch:
        ystretchbottom = pb_ymin_oycent

    if pb_ymax_oycent < ystretch and pb_ymin_oycent > ystretch:
        ystretchtop = pb_ymax_oycent
        ystretchdiff = ystretch - ystretchtop
        ystretchbottom = (ystretch + ystretchdiff) if pb_ymin_oycent > (ystretch +
ystretchdiff) else pb_ymin_oycent
    elif pb_ymax_oycent < ystretch and pb_ymin_oycent <= ystretch:
        ystretchtop = pb_ymax_oycent

    bxpixels = int((xmax - xmin) / xres)
    bypixels = int((ymax - ymin) / yres)
    randxmax = (bxpixels - xpixels) if (bxpixels - xpixels) > 0 else 1
    randymax = (bypixels - ypixels) if (bypixels - ypixels) > 0 else 1

    randx = np.random.randint(0, randxmax)
    randy = np.random.randint(0, randymax)

    cropxmin = xmin + (randx * xres)
    cropxmax = xmin + ((randx + xpixels) * xres)
    cropymin = ymax - ((randy + ypixels) * yres)
    croppymax = ymax - (randy * yres)

    crop_bbox = [Polygon([[cropxmin, cropymin], [cropxmax, cropymin], [cropxmax,
croppymax], [cropxmin, croppymax]])]
    crop_bbox_gdf = gpd.GeoDataFrame(crop_bbox, columns=["geometry"],
crs=sat_crs)

    if plot_box:
        fig, ax = plt.subplots(figsize=(10, 6))
        sat_bbox_gdf.plot(color='orange', ax=ax)
        oil_bbox_gdf.plot(color='red', ax=ax)
        crop_bbox_gdf.plot(color='cyan', alpha=0.5, ax=ax)
        plt.show()

    return crop_bbox_gdf

except Exception as error:
    print(f"\tError in get_simple_crop_bbox: {error}")
    raise Exception(f"\tError in get_simple_crop_bbox: {error}")

```

A4 Створення композиту Oil Spill Index

```
def get_rgb_composite_image(base_dir, image_path, band_ratio_name):
    try:
        image_xarray = rxr.open_rasterio(os.path.join(base_dir, image_path))
        if band_ratio_name == 'OSI':
            bands_3 = image_xarray.sel(band=[3])
            bands_4 = image_xarray.sel(band=[4])
            bands_2 = image_xarray.sel(band=[2])

            bands_3['band'] = ('band', [22])
            bands_4['band'] = ('band', [22])
            bands_2['band'] = ('band', [22])

            rgb_combo = (bands_3 + bands_4) / bands_2
            rgb_combo_norm = np.floor((rgb_combo - rgb_combo.min(dim=['x', 'y'])) * 255
/ (rgb_combo.max(dim=['x', 'y']) - rgb_combo.min(dim=['x', 'y'])))
```