

уМіністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Сегментація об'єктів у супутникових зображеннях. Використання штучного інтелекту для автоматичної класифікації ділянок на супутникових знімках
(тема)

Виконав:
здобувач другого року навчання,
групи СШМ-23-2

Володимир Ярцев
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)

Керівник проф. Кирило Смеляков
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____
(підпис)

Олег ЗОЛОТУХІН
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

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

Освітня програма _____ Системи штучного інтелекту _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Ярцеву Володимирі Романовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Сегментація об'єктів у супутникових зображеннях. Використання штучного інтелекту для автоматичної класифікації ділянок на супутникових знімках

затверджена наказом університету від 21 квітня 2025 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 2025 р.

3. Вихідні дані до роботи Науково-технічні публікації, матеріали з міжнародного семінару CMIS-2025, інтернет-джерела з методів сегментації супутникових зображень, набір даних Poland Aerial Dataset та API-документація Google Maps Static API

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз методів сегментації зображень _____

2) Експерименти та результати _____

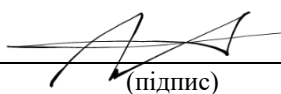
3) Програмна реалізація _____

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Строк / терміни виконання етапів роботи | Примітка |
|----|---|---|----------|
| 1 | Отримання завдання на кваліфікаційну роботу | 21.04.2025 | виконано |
| 2 | Ознайомлення з предметною галуззю | 22.04.2025 | виконано |
| 3 | Визначення структури роботи, пошук та вивчення літератури | 25.04.2025 | виконано |
| 4 | Формування постановки завдання | 28.04.2025 | виконано |
| 5 | Експериментальне моделювання та навчання моделі | 05.05.2025 | виконано |
| 6 | Написання пояснювальної записки | 16.05.2025 | виконано |
| 7 | Перевірка на академічний плагіат | 17.05.2025 | виконано |
| 8 | Нормоконтроль | 18.05.2025 | виконано |
| 9 | Підготовка презентації та доповіді | 20.05.2025 | виконано |
| 10 | Попередній захист | 22.05.2025 | виконано |
| 11 | Рецензування | 03.06.2025 | виконано |
| 12 | Захист перед ЕК | 06.06.2025 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____



(підпис)

Керівник роботи _____

(підпис)

проф. Кирило Смеляков

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 62 с., 25 рис., 3 табл., 1 дод., 31 джерело.

АВТОМАТИЧНА КЛАСИФІКАЦІЯ, ГЛИБОКЕ НАВЧАННЯ, ДИСТАНЦІЙНЕ ЗОНДУВАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, СЕГМЕНТАЦІЯ ОБ'ЄКТІВ, СУПУТНИКОВІ ЗНІМКИ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – процес автоматичної сегментації об'єктів на супутникових зображеннях за допомогою методів штучного інтелекту.

Предмет дослідження – методи та архітектури глибокого навчання, зокрема згорткові нейронні мережі, що застосовуються для семантичної сегментації супутникових знімків.

Метою даної роботи є виявлення найефективніших методів сегментації, розробка та апробація методів сегментації об'єктів на супутникових знімках з використанням алгоритмів штучного інтелекту для автоматичної класифікації ділянок. Для досягнення поставленої мети передбачено вирішення наступних завдань:

- аналіз сучасних методів сегментації та класифікації супутникових зображень;
- розробка алгоритмічної бази для попередньої обробки та сегментації зображень;
- розробка та навчання моделі штучного інтелекту для автоматичної класифікації ділянок;
- експериментальна перевірка ефективності запропонованих методів з використанням реальних даних.

ABSTRACT

Master's thesis contains: 62 pp., 25 fig., 3 tabl., 1 ann., 31 references.

ARTIFICIAL INTELLIGENCE, AUTOMATIC CLASSIFICATION, CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, MACHINE LEARNING, OBJECT SEGMENTATION, REMOTE SENSING, SATELLITE IMAGERY.

Object tracking is the process of automatic segmentation of objects on satellite images using artificial intelligence methods.

The subject of research is the methods and architecture of deep learning, focusing on neural networks that are used for semantic segmentation of satellite images.

The purpose of this work is to identify the most effective segmentation methods, develop and test object segmentation methods on satellite images using artificial intelligence algorithms for automatic classification of plots. To achieve the goal set, the following steps have been given:

- analysis of current methods for segmentation and classification of satellite images;
- development of an algorithmic basis for advanced processing and image segmentation;
- development and development of a model of piece intelligence for automatic classification of plots;
- experimental verification of the effectiveness of the proposed methods against real data.

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 8 |
| Вступ..... | 9 |
| 1 Аналіз методів сегментації зображень | 11 |
| 1.1 Задачі сегментації..... | 11 |
| 1.1.1 Визначення сегментації зображень..... | 11 |
| 1.1.2 Семантична сегментація..... | 11 |
| 1.1.3 Сегментація об'єктів..... | 12 |
| 1.1.4 Паноптична сегментація | 13 |
| 1.1.5 Галузеве використання..... | 14 |
| 1.2 Метрики оцінювання точності сегментації..... | 16 |
| 1.2.1 Піксельна точність..... | 16 |
| 1.2.2 Індекс Жаккара..... | 17 |
| 1.2.3 Коефіцієнт подібності Дайса | 19 |
| 1.3 Архітектурні моделі конволюційних нейронних мереж..... | 21 |
| 1.3.1 Архітектура U-Net..... | 21 |
| 1.3.2 Архітектура DeepLab..... | 23 |
| 1.3.3 Архітектура EfficientNet..... | 25 |
| 1.4 Аналіз існуючих рішень | 27 |
| 1.4.1 Google Earth Engine..... | 27 |
| 1.4.2 Sentinel Hub..... | 28 |
| 1.4.3 Descartes Labs | 30 |
| 1.4.4 Висновок щодо існуючих систем..... | 31 |
| 2 Експерименти та результати | 33 |
| 2.1 Підготовка середовища перед експериментами | 33 |
| 2.2 Навчання моделей та результати експериментів | 34 |
| 2.3 Висновки щодо експериментів | 36 |
| 3 Програмна реалізація..... | 40 |
| 3.1 Вибір мови програмування та технологій | 40 |

| | |
|--|----|
| 3.2 Вибір та підготовка датасету | 41 |
| 3.3 Архітектура веб-додатку | 42 |
| 3.4 Розробка серверної частини (Backend) | 45 |
| 3.5 Розробка клієнтської частини (Frontend)..... | 46 |
| 3.6 Реалізація моделі сегментації | 47 |
| 3.7 Інтеграція моделі в веб-додаток | 49 |
| 3.8 Візуалізація результатів..... | 50 |
| 3.9 Тестування системи | 52 |
| 3.10 Висновки щодо розробленої системи | 54 |
| Висновки | 56 |
| Перелік джерел посилання | 57 |
| Додаток А Відомість кваліфікаційної роботи | 62 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – інтерфейс прикладного програмування;

Argmax – функція, яка обирає індекс максимального значення;

ASPP – Atrous Spatial Pyramid Pooling – багаторівнева контекстна агрегація;

Base64 – спосіб кодування двійкових даних у текст;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

DSC – Dice Similarity Coefficient – коефіцієнт подібності Дайса;

Focal Loss – функція втрат для задач з дисбалансом класів;

HTTP – HyperText Transfer Protocol – протокол передачі гіпертексту;

Inference – виконання прогнозу моделі на нових даних;

IoU – Intersection over Union – індекс Жаккара, метрика якості сегментації;

JSON – JavaScript Object Notation – формат обміну даними;

mPA – mean Pixel Accuracy – середня піксельна точність;

ONNX – Open Neural Network Exchange – відкритий формат для нейронних мереж;

PA – Pixel Accuracy – піксельна точність;

ReLU – Rectified Linear Unit – активаційна функція;

RGB – Red Green Blue – формат кольорового зображення;

SiLU / Swish – Smooth ReLU – плавна активаційна функція;

SPA – Single Page Application – односторінковий додаток;

U-Net – архітектура згорткової нейронної мережі для сегментації зображень.

ВСТУП

У сучасному світі інформаційних технологій та високих швидкостей обробки даних супутникові знімки відіграють ключову роль у моніторингу навколишнього середовища, плануванні міст, сільському господарстві, екологічних дослідженнях та численних інших сферах. Розвиток дистанційного зондування Землі дозволив отримувати великі обсяги даних, що містять важливу інформацію про стан поверхні планети. Проте ефективне використання цих даних значною мірою залежить від можливості правильної інтерпретації зображень та їх автоматичної обробки.

Сегментація об'єктів є однією з основних задач обробки супутникових знімків, яка передбачає поділ зображення на окремі, однорідні області, кожна з яких має певну інтерпретацію в контексті аналізу. Завдяки цьому можна виділити природні ландшафти, міські території, водні об'єкти, сільськогосподарські угіддя та інші структурні елементи, що полегшує їх подальшу класифікацію та аналіз. Проте традиційні методи сегментації часто стикаються з труднощами через високу складність природних ландшафтів, різноманітність форм, текстур та змін освітлення.

Використання штучного інтелекту, зокрема сучасних алгоритмів машинного навчання та глибокого навчання, стало проривом у вирішенні цих проблем. Інтелектуальні системи здатні адаптивно аналізувати великі масиви даних, знаходити закономірності та виконувати класифікацію з високою точністю навіть за наявності шуму та неповноти даних. Застосування нейронних мереж, таких як згорткові нейронні мережі (CNN) та їх модифікацій, дозволяє автоматично навчатися розпізнаванню характерних ознак об'єктів на зображеннях, що істотно підвищує ефективність аналізу супутникових знімків.

Актуальність даної теми обумовлена не тільки швидким розвитком технологій дистанційного зондування, але й необхідністю оперативного прийняття рішень у різних галузях. Сучасні суспільні виклики, зокрема

зміни клімату, зростання міського населення та потреба у раціональному використанні природних ресурсів, потребують від науковців і практиків створення високоточних, автоматизованих інструментів для аналізу просторових даних.

При підготовці кваліфікаційної роботи була підготовлена та опублікована наукова робота: «Моделі штучного інтелекту для автоматичної класифікації об'єктів на супутникових знімках» у межах «Восьмого міжнародного семінара з комп'ютерного моделювання та інтелектуальних систем (CMIS-2025)», 5 травня 2025 року, м. Запоріжжя [1].

1 АНАЛІЗ МЕТОДІВ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

1.1 Задачі сегментації

1.1.1 Визначення сегментації зображень

Сегментація зображень – це ключова техніка комп’ютерного зору та обробки зображень, що використовується для поділу зображення на окремі сегменти або області, які мають схожі характеристики. Основна мета сегментації полягає в тому, щоб спростити аналіз зображення та зробити його більш структурованим для подальшої обробки.

У загальному випадку сегментація може бути розглянута у двох аспектах: розділення об’єктів та сутностей. Об’єкти – це злічувані предмети, які мають чітко визначені контури та форми (наприклад, люди, автомобілі, будівлі, тварини). Сутності, у свою чергу, є аморфними та незлічуваними (наприклад, вода, небо, рослинність) [2]. Ця класифікація є важливою для розуміння різних типів алгоритмів сегментації, оскільки різні методи по-різному справляються з такими категоріями.

1.1.2 Семантична сегментація

Семантична сегментація – техніка комп’ютерного зору, яка передбачає позначення та сегментування кожного пікселя зображення на основі його семантичного вмісту. Ця технологія знаходить застосування в таких галузях, як аналіз супутникових знімків, автономне керування транспортом та медична діагностика. Традиційні алгоритми сегментації мають певні обмеження, зокрема щодо точності визначення меж об’єктів і виявлення дрібних або складних елементів [3]. Впровадження глибокого навчання значно покращило цей процес, зробивши його більш ефективним і точним.

Одним із поширених підходів у семантичній сегментації є архітектура, що поєднує кодер і декодер. Кодер виконує стискання вхідного зображення, поступово зменшуючи його розмір і виокремлюючи важливі ознаки за допомогою згорткових шарів. Декодер, у свою чергу, використовує ці ознаки для відновлення структури зображення, присвоюючи кожному пікселю відповідну мітку класу. Існують різні варіанти цього підходу, які передбачають різні способи з'єднання між рівнями кодування та декодування для підвищення точності сегментації [4]. На рисунку 1.1 наведено демонстрацію роботи нейронної мережі для семантичної сегментації в автономному водінні.



Рисунок 1.1 – Демонстрація роботи сегментації в автономному водінні

1.1.3 Сегментація об'єктів

Сегментація об'єктів (або сегментація екземплярів) є розширенням семантичної сегментації, що дозволяє не лише класифікувати області, а й

розрізняти окремі об'єкти одного класу. Наприклад, на знімку може бути кілька будівель, і завдання сегментації об'єктів полягає в тому, щоб кожену будівлю виділити як окрему сутність.

Цей підхід широко використовується для моніторингу міських територій, ідентифікації транспортних засобів на дорогах, а також в екологічних дослідженнях. Популярними методами сегментації об'єктів є Mask R-CNN та Faster R-CNN [5]. Ці алгоритми дозволяють створювати точні маски для кожного окремого об'єкта, що значно покращує автоматичний аналіз знімків. На рисунку 1.2 наведено демонстрацію роботи нейронної мережі для сегментації об'єктів.

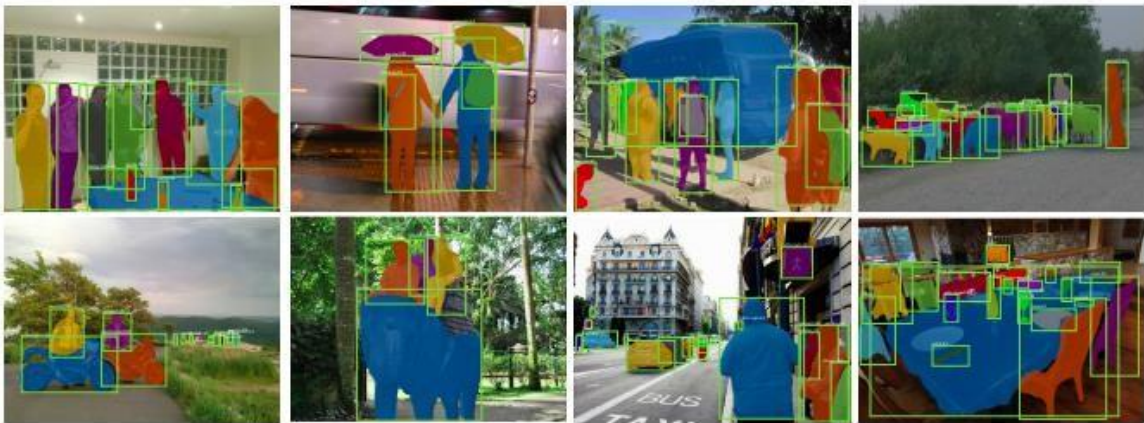


Рисунок 1.2 – Демонстрація роботи сегментації об'єктів

1.1.4 Паноптична сегментація

Паноптична сегментація об'єднує підходи семантичної та об'єктної сегментації, дозволяючи одночасно класифікувати сутності та розпізнавати окремі об'єкти. Вона надає кожному пікселю зображення два атрибути: семантичну мітку (тип об'єкта) та ідентифікатор екземпляра (унікальний ID для кожного окремого об'єкта).

Паноптична сегментація особливо корисна у випадках, коли необхідно ідентифікувати складні сцени, що містять як окремі об'єкти, так і незліченні сутності. Наприклад, при аналізі міських ландшафтів вона

дозволяє виділяти дороги, тротуари, будівлі, транспортні засоби та пішоходів одночасно [6]. Одним з ключових викликів цього підходу є необхідність розв'язання проблеми перекриття пікселів між об'єктами.

Сучасні моделі паноптичної сегментації базуються на методі Mask R-CNN та його вдосконалених версіях, таких як UPSNet, FPSNet, EPSNet та VPSNet. Вони поєднують переваги глибокого навчання та спеціалізованих архітектур, що дозволяє досягти високої точності розпізнавання об'єктів та сутностей на супутникових знімках [7]. На рисунку 1.3 наведено демонстрацію роботи нейронної мережі для паноптичної сегментації.



Рисунок 1.3 – Демонстрацію роботи паноптичної сегментації

1.1.5 Галузеве використання

Дані методи сегментації зображень широко застосовуються у сфері комп'ютерного зору та аналізу візуальної інформації. Їх комбіноване використання відкриває можливості для створення інноваційних рішень, що значно розширюють здатність людини до обробки складних даних.

Одним із ключових напрямків застосування є автономний транспорт. Тривимірна семантична сегментація дозволяє самокерованим автомобілям

детально аналізувати навколишнє середовище, розпізнавати об'єкти на дорозі та ухвалювати рішення в реальному часі [8]. Сегментація об'єктів, своєю чергою, сприяє точному визначенню кожного елемента сцени, що необхідно для обчислення безпечної дистанції, швидкості руху та прогнозування траєкторій.

В медицині ці технології використовуються для діагностики за допомогою МРТ, КТ та рентгенографії. Завдяки алгоритмам сегментації можна автоматично визначати патологічні зміни, такі як пухлини, внутрішні кровотечі чи дегенеративні процеси, що покращує точність діагностики та прискорює ухвалення клінічних рішень.

Сегментація зображень також відіграє важливу роль у геоінформаційних системах. Вона використовується для аналізу супутникових та аерофотознімків, допомагаючи у картографуванні місцевості [9]. Ці методи дозволяють ідентифікувати природні та штучні об'єкти, зокрема річки, ліси, дороги, міські забудови та сільськогосподарські угіддя, що важливо для моніторингу змін навколишнього середовища.

Паноптична сегментація, що поєднує семантичну та інстанс-сегментацію, забезпечує ще вищий рівень деталізації. Вона використовується в розширеному аналізі відеоспостереження, анотації даних для навчання штучного інтелекту, обробці зображень, отриманих з безпілотних літальних апаратів, а також для підрахунку людей у натовпі [10]. Ця технологія покращує якість прогнозування, підвищує рівень автоматизації та дозволяє ефективно обробляти великі обсяги візуальних даних у реальному часі.

Окрім традиційних сфер застосування, сегментація об'єктів поступово інтегрується у сферу доповненої та віртуальної реальності. Завдяки їй можна створювати реалістичні цифрові середовища, в яких об'єкти точно відокремлюються від фону та взаємодіють з користувачем у режимі реального часу. Це відкриває нові можливості для ігор, симуляцій та

навчальних платформ. Також значний потенціал мають розумні міста, де сегментація зображень сприяє аналізу транспортних потоків, управлінню інфраструктурою та підвищенню безпеки завдяки автоматизованому розпізнаванню об'єктів у міському просторі [11].

1.2 Метрики оцінювання точності сегментації

Оцінка якості роботи алгоритму сегментації є ключовим етапом аналізу його ефективності. Використання відповідних метрик дозволяє об'єктивно визначити, наскільки точно модель виконує поділ зображення на області та чи відповідає отриманий результат очікуваному. Оскільки різні алгоритми можуть мати свої особливості та застосовуватись у різних сферах, важливо обирати метрики, що враховують специфіку конкретного завдання. Для цього використовуються як стандартні статистичні показники, так і спеціалізовані метрики, що оцінюють схожість між передбаченим та еталонним сегментованим зображенням.

1.2.1 Піксельна точність

Піксельна точність (Pixel Accuracy – PA) є однією з базових метрик семантичної сегментації, яка визначає частку правильно класифікованих пікселів від загальної кількості пікселів у зображенні [12]. Вона показує, наскільки точно алгоритм розподіляє кожен піксель між визначеними класами:

$$PA = \frac{\sum_{j=1}^k n_{jj}}{\sum_{j=1}^k t_j}, \quad (1.1)$$

де n_{jj} – це сумарна кількість пікселів, які були правильно віднесені до класу j .

Іншими словами, n_{jj} відповідає числу істинно позитивних (True Positive) результатів для даного класу j . t_j – це загальна кількість пікселів, що належать класу j згідно з розміткою.

Оскільки в задачах семантичної сегментації можуть бути присутні декілька класів, усереднена піксельна точність (mean Pixel Accuracy – mPA) дозволяє оцінити середнє значення точності класифікації для всіх класів [13]. Це робить її більш придатною для аналізу моделей, які працюють із різноманітними класами об'єктів:

$$mPA = \frac{1}{k} \sum_{j=1}^k \frac{n_{jj}}{t_j}. \quad (1.2)$$

Попри те, що PA та mPA є зрозумілими та легкоінтерпретованими метриками, вони не завжди точно відображають реальну ефективність алгоритму. Висока точність класифікації пікселів не завжди свідчить про якісну сегментацію, особливо якщо набір даних містить дисбаланс класів [14]. Наприклад, у випадках, коли один із класів (найчастіше фон) займає більшу частину зображення, модель може досягати високого PA, просто правильно класифікуючи цей домінуючий клас, ігноруючи менш представлені об'єкти. Тому, для об'єктивного оцінювання роботи алгоритму важливо використовувати додаткові метрики, такі як IoU (Intersection over Union), які враховують точність та повноту розпізнавання кожного класу.

1.2.2 Індекс Жаккара

Індекс Жаккара (Intersection over Union – IoU) – одна з основних метрик, що використовується для оцінки якості алгоритмів семантичної сегментації. Ця метрика відображає ступінь збігу між передбаченою сегментацією та еталонною розміткою, дозволяючи визначити, наскільки

точно модель відокремлює об'єкти на зображенні [15]. Індекс Жаккара наведено на рисунку 1.4.

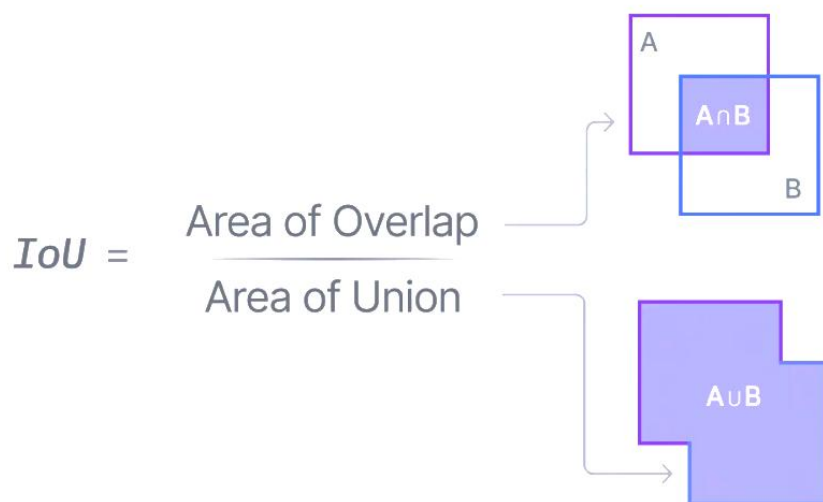


Рисунок 1.4 – Індекс Жаккара (IoU)

IoU розраховується як відношення площі перетину (перекриття) між прогнозованою маскою та реальною розміткою до загальної площі їх об'єднання [16]. Значення цієї метрики варіюється від 0 до 1 (або від 0% до 100%):

- 0 означає повну відсутність збігу, коли модель не ідентифікувала об'єкт правильно;

- 1 вказує на ідеальне співпадіння, коли прогнозована сегментація повністю збігається з істинною маскою.

Якщо виразити це математично, то отримаємо:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}, \quad (1.3)$$

де A – це прогнозований результат сегментації, отриманий від моделі;

B – це правильний результат сегментації.

Формально для бінарної або багатокласової сегментації середнє значення IoU визначається шляхом підрахунку IoU для кожного класу та подальшого усереднення, подібно до середньої піксельної точності (mPA).

Попри свою популярність, IoU має певні обмеження. Наприклад, у випадках, коли класи в наборі даних сильно дисбалансовані, ця метрика може не завжди коректно відображати ефективність сегментації для рідко представлених об'єктів [17]. Саме тому її часто доповнюють такими показниками, як Dice Coefficient (F1-Score для сегментації), який більшою мірою враховує як точність, так і повноту передбаченої сегментації.

Приклад сегментації методом індекса Жаккара наведено на рисунку 1.5.



Рисунок 1.5 – Приклад роботи IoU

1.2.3 Коефіцієнт подібності Дайса

Для завдань бінарної сегментації коефіцієнт подібності Дайса (Dice Similarity Coefficient, DSC) або F1-міра є мірою схожості між двома сегментаційними масками. Він визначається як подвоєне значення площі їхнього перетину, поділене на загальну кількість пікселів у прогнозованій та еталонній масках [18]. Коефіцієнт подібності Дайса наведено на рисунку 1.6.

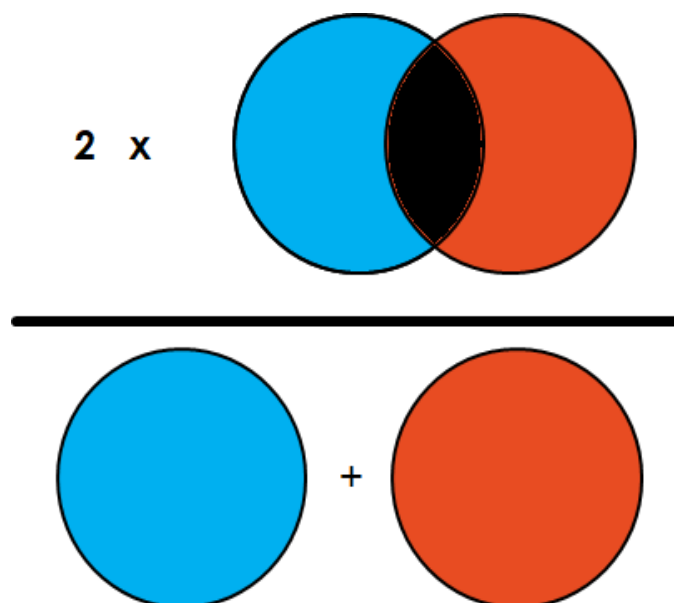


Рисунок 1.6 – Коефіцієнт подібності Дайса

В математичній нотації це матиме наступний вигляд:

$$\frac{2 \cdot |x \cap y|}{|x| + |y|} \quad (1.4)$$

де X – сегментаційна маска, отримана в результаті роботи моделі;

Y – референсна (еталонна) сегментаційна маска.

Подібно до індексу Жаккара (IoU), коефіцієнт Дайса має діапазон значень від 0 до 1 (або від 0% до 100%).

– 0 означає повну відсутність збігу, тобто модель не змогла правильно виділити об'єкт;

– 1 вказує на ідеальне співпадіння, коли сегментована область точно повторює розмітку.

Хоча обидві метрики IoU та DSC – позитивно корелюють і зазвичай дають схожі оцінки, вони мають певні відмінності. IoU є більш суворою метрикою, оскільки сильніше штрафує погано класифіковані випадки [19]. Це означає, що навіть якщо обидві метрики показують невдалу сегментацію на певних зображеннях, IoU може оцінювати такі помилки більш жорстко.

Натомість коефіцієнт Дайса є більш згладженою мірою, оскільки менш чутливий до окремих випадків поганої класифікації. Він частіше використовується для оцінки загальної продуктивності моделі, тоді як IoU більше зосереджується на деталях та якості прогнозу в складних ситуаціях [20].

У медичних застосуваннях, зокрема при аналізі МРТ та КТ-знімків, коефіцієнт Дайса є однією з ключових метрик, оскільки він дозволяє точно оцінювати рівень збігу між передбаченою та реальною анатомічною структурою.

1.3 Архітектурні моделі конволюційних нейронних мереж

1.3.1 Архітектура U-Net

U-Net є однією з найпопулярніших архітектур згорткових нейронних мереж. Її основна особливість полягає в тому, що вона не лише класифікує зображення загалом, а й виконує піксельну сегментацію, тобто створює маску, яка розподіляє об'єкти на відповідні класи. Структура мережі включає два основні компоненти: енкодер (стискаючий шлях), який аналізує контекст, і декодер (розширюючий шлях), що відповідає за точне відновлення просторової інформації.

Однією з головних переваг U-Net є здатність ефективно працювати з обмеженими наборами даних. Модель була успішно застосована у змаганні ISBI, де перевершила попередні методи, зокрема згорткові мережі зі скользячим вікном, для задачі сегментації нейронних структур в електронно-мікроскопічних зображеннях [21]. Завдяки своїй універсальності та потужності, U-Net також досягла найкращих результатів у конкурсі ISBI 2015 з відстеження клітин на мікроскопічних знімках, таких як фазовий контраст і DIC, значно випереджаючи конкурентів.

Окрім високої точності, ще однією важливою характеристикою U-Net є її швидкодія: сегментація зображення розміром 512×512 пікселів займає всього кілька мілісекунд на сучасному графічному процесорі [22]. Це робить її придатною для широкого спектра застосувань, особливо в медичній діагностиці, де вона використовується для аналізу МРТ, КТ та гістологічних знімків.

U-Net довела свою ефективність не лише в біомедичних задачах, а й у різних інших сферах, зокрема в дистанційному зондуванні, аналізі супутникових знімків, автоматичному розпізнаванні об'єктів на промислових і наукових зображеннях [23]. Її здатність працювати з обмеженою кількістю навчальних прикладів робить цю архітектуру однією з найкращих для завдань сегментації в умовах недостатньої кількості даних.

Приклад архітектури U-Net наведено на рисунку 1.7.

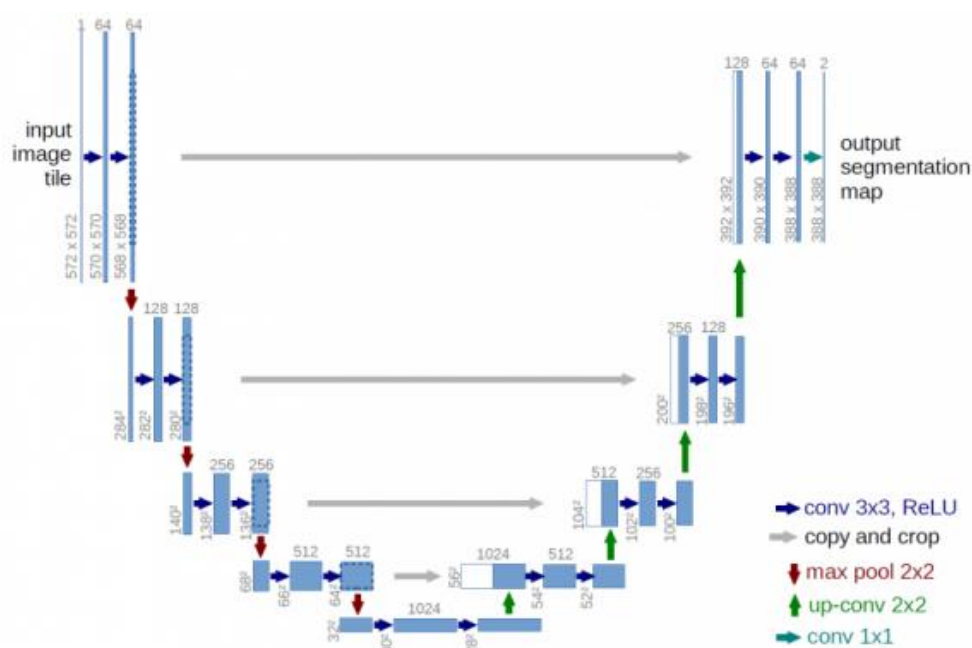


Рисунок 1.7 – Архітектура U-Net

На рисунку кожен синій квадрат відповідає властивості багатоканальної карти. Кількість каналів наведено у верхній частині квадрата. Розмір x-y наведено в нижньому лівому краю квадрата. Білі

квадрати являють собою копію властивості карти. Стрілки означають різні операції [24].

Приклад роботи U-Net було взято сегментації клітин на зображеннях з оптичним мікроскопом продемонстровано на рисунку 1.8.

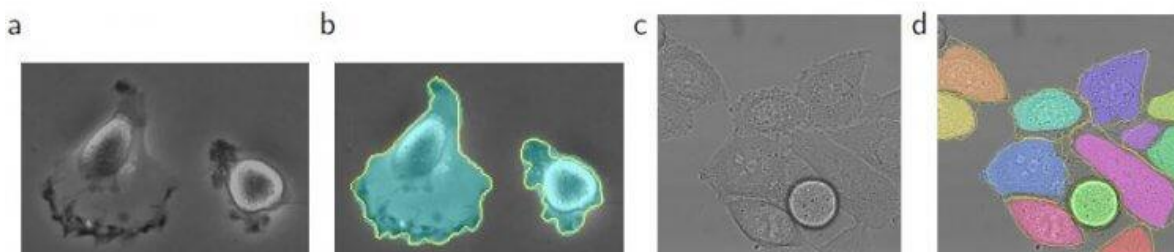


Рисунок 1.8 – (а) Частина вхідного зображення датасету PhC-U373. (б) Результат сегментації (блакитна маска) та мітка (жовта межа). (с) Вхідне зображення датасету DIC-HeLa. (д) Результат сегментації (кольорові маски) та мітки (жовта рамка)

1.3.2 Архітектура DeepLab

DeepLab – сімейство архітектур згорткових нейронних мереж (CNN), призначених для семантичної сегментації в комп'ютерному зорі. Ці моделі відомі своєю здатністю захоплювати дрібні деталі та виконувати семантичну сегментацію зображень із високою роздільною здатністю.

Модуль кодера в архітектурі DeepLab відповідає за обробку багатомасштабної контекстної інформації шляхом застосування розширеної згортки (dilated convolution) на різних рівнях. Це дозволяє зберігати високу деталізацію об'єктів, покращуючи їх сегментацію навіть у складних умовах [25]. Декодер, у свою чергу, виконує уточнення меж об'єктів, що підвищує точність сегментації, особливо для дрібних деталей.

Архітектуру DeepLab наведено на рисунку 1.9. Приклад роботи DeepLab наведено на рисунку 1.10.

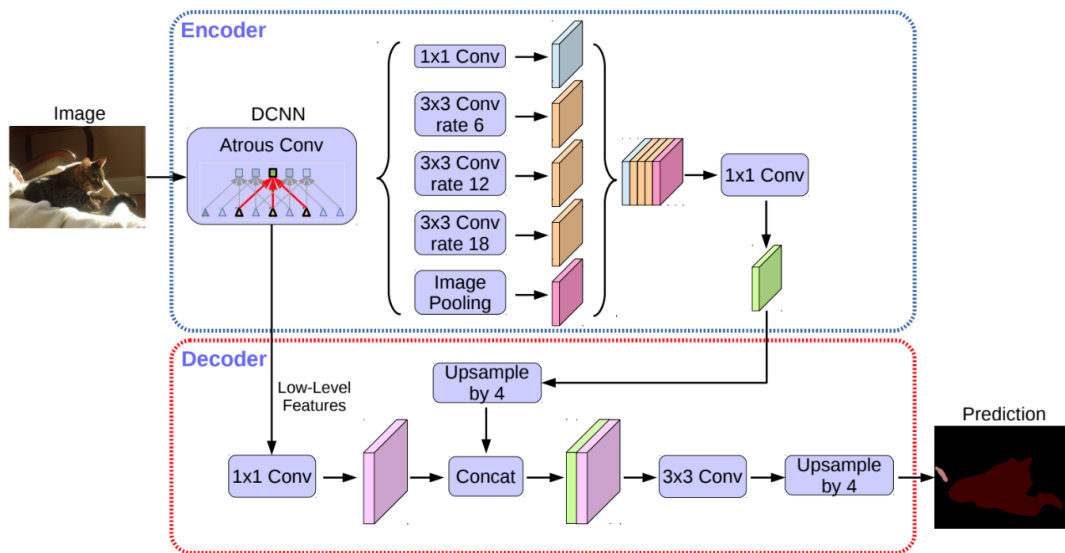


Рисунок 1.9 – Архітектура Deeplab

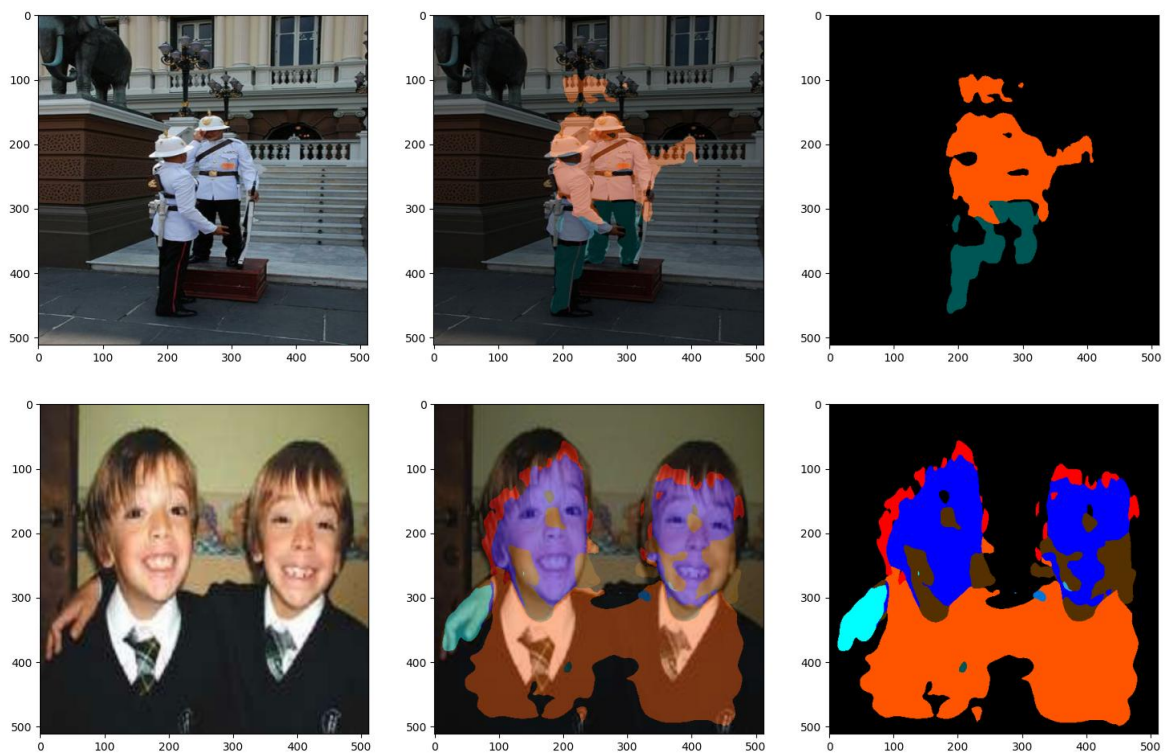


Рисунок 1.10 – Архітектура Deeplab

Однією з ключових особливостей Deeplab є використання розширеної згортки, яка дозволяє збільшувати поле зору мережі без збільшення кількості параметрів або обчислювальної складності [26]. Це

забезпечує ефективне захоплення контекстної інформації та дає змогу отримувати високоякісні карти функцій, що є важливим фактором для точного розподілу класів у семантичній сегментації.

Додатково DeepLab використовує підхід Atrous Spatial Pyramid Pooling (ASPP), який дозволяє обробляти інформацію з різними масштабами контексту [27]. Це сприяє поліпшенню розпізнавання об'єктів незалежно від їх розміру на зображенні.

1.3.3 Архітектура EfficientNet

EfficientNet – це сімейство згорткових нейронних мереж, розроблене компанією Google Brain, яке поєднує високу точність класифікації з ефективним використанням обчислювальних ресурсів. Його ключовою особливістю є метод масштабування мережі Compound Scaling, який одночасно змінює три параметри: глибину (depth), ширину (width) та роздільну здатність (resolution) замість того, щоб збільшувати лише один з них [28]. Такий підхід дозволяє досягти оптимального балансу між продуктивністю та витратами обчислювальних ресурсів.

Архітектура EfficientNet базується на MobileNetV2 та використовує інверсні резидуальні блоки (Inverted Residual Blocks) і глибинні згортки (Depthwise Convolutions), що зменшують кількість параметрів без втрати продуктивності. Крім того, для активації нейронів застосовується функція Swish (SiLU), яка сприяє кращому передаванню градієнтів та підвищує точність моделі порівняно з класичною ReLU [29]. Приклад архітектури EfficientNet представлено на рисунку 1.11.

Завдяки своїй ефективності ця архітектура широко застосовується у завданнях класифікації зображень, семантичній сегментації та розпізнавання об'єктів, особливо у випадках, коли важливо мінімізувати витрати ресурсів без втрати продуктивності [30].

Приклади роботи EfficientNet представлено на рисунку 1.12.

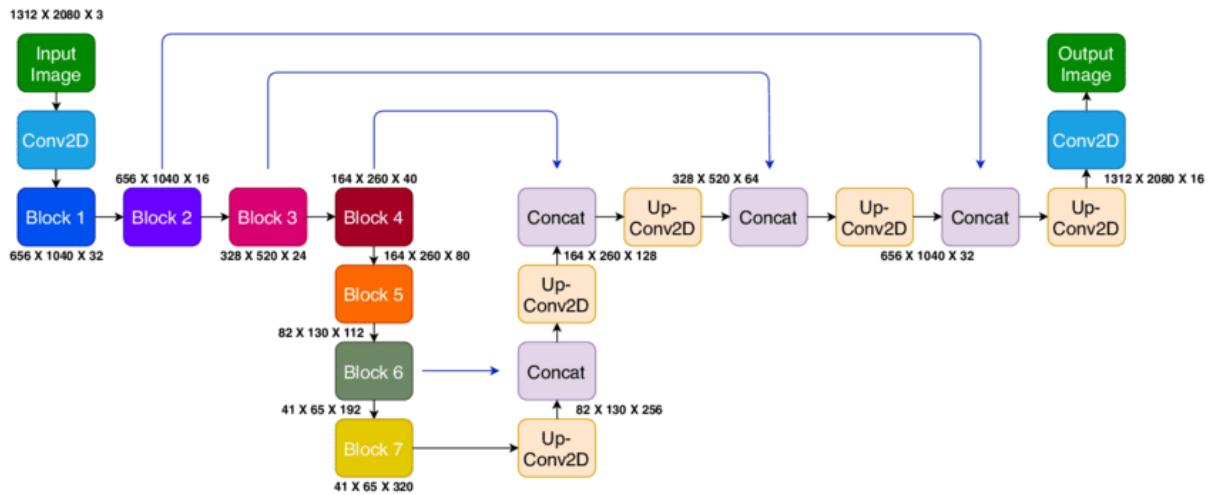


Рисунок 1.11 – Архітектура EfficientUNet із фреймворком EfficientNet-B0 для семантичної сегментації

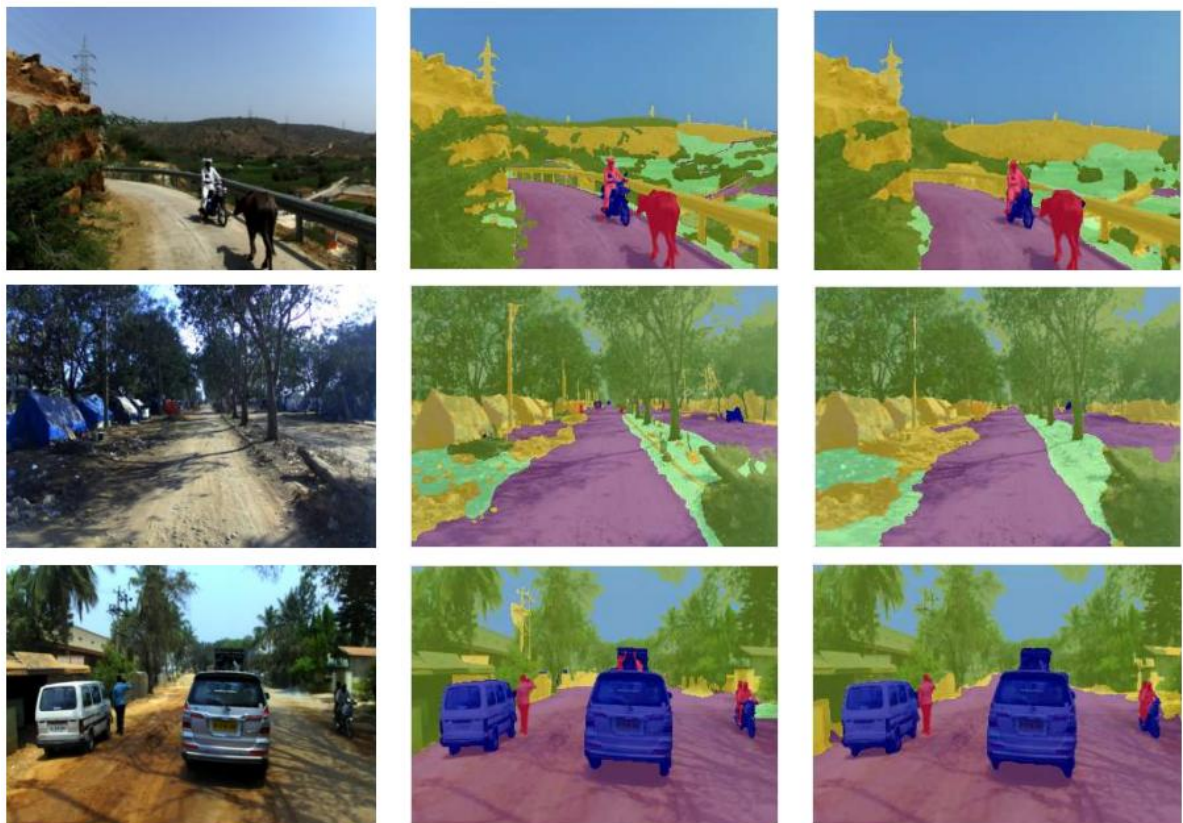


Рисунок 1.12 – Результат семантичної сегментації EfficientNet.

Перший стовпець показує вхідні зображення, що відображають різні сценарії з неструктурованого середовища. Другий і третій стовпці

показують базову істину та прогнозовану карту сегментації відповідно, де різні кольори позначають різні класи

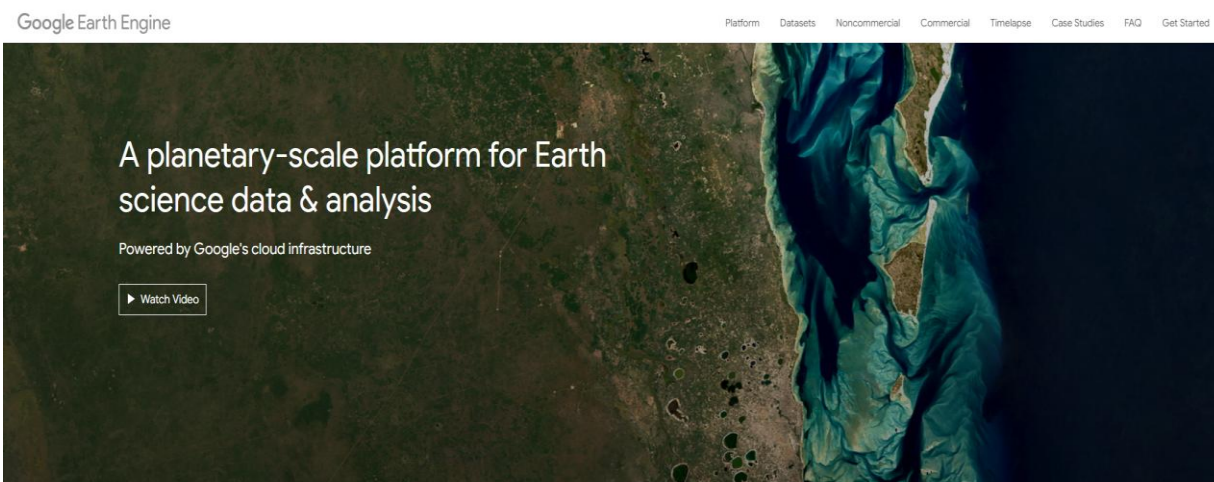
EfficientUNet демонструє хороше узагальнення на складних ландшафтах та зменшує час обробки, що робить її придатною для застосувань у реальному часі або в умовах обмежених обчислювальних потужностей. Крім того, її модульна структура дозволяє легко адаптувати модель до різних кількостей класів та типів даних, що забезпечує гнучкість у розв'язанні практичних задач, пов'язаних з аналізом земного покриття, моніторингом змін територій і виявленням об'єктів інфраструктури.

EfficientNets також добре передає та досягає найсучаснішої точності на CIFAR-100 (91,7%), Flowers (98,8%) та 3 інших наборах даних для навчання передачі з на порядок меншою кількістю параметрів [31]. Такий високий рівень узагальнення досягається завдяки збалансованій архітектурі EfficientNet, яка масштабується одночасно по глибині, ширині та роздільній здатності. Це дозволяє досягати кращих результатів при меншому обсязі обчислень порівняно з класичними архітектурами. У контексті семантичної сегментації це особливо важливо, оскільки модель може обробляти великі обсяги зображень зі складною структурою (наприклад, міське середовище чи природні ландшафти) без значного навантаження на апаратні ресурси.

1.4 Аналіз існуючих рішень

1.4.1 Google Earth Engine

Google Earth Engine – це платформа для планетарного аналізу геопросторових даних, що поєднує величезну бібліотеку супутникових знімків (як-от дані від Landsat, Sentinel та інших) із потужними інструментами обробки даних та можливостями машинного навчання. На рисунку 1.13 наведено сервіс Google Earth Engine.



Meet Earth Engine

Google Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities. Scientists, researchers, and developers use Earth Engine to detect changes, map trends, and quantify differences on the Earth's surface. Earth Engine is now available for commercial use, and remains free for academic and research use.

Рисунок 1.13 – Сервіс Google Earth Engine

Переваги:

- доступ до великих наборів даних і регулярних оновлень супутникових зображень;
- інтеграція з API на javascript та python, що дозволяє розробляти власні алгоритми сегментації та класифікації;
- безкоштовний доступ для наукових досліджень та некомерційного використання.

Недоліки:

- потребує глибоких знань з програмування та роботи з API;
- інтерфейс може здатися складним для новачків;
- обмеження на обчислювальні ресурси для великих проектів.

1.4.2 Sentinel Hub

Sentinel Hub – це хмарна платформа для обробки та доступу до супутникових даних, зокрема від місій Sentinel Європейського космічного

агентства. Вона надає інструменти для візуалізації, аналізу та обробки зображень через веб-інтерфейс та API. На рисунку 1.14 наведено сервіс Sentinel Hub.

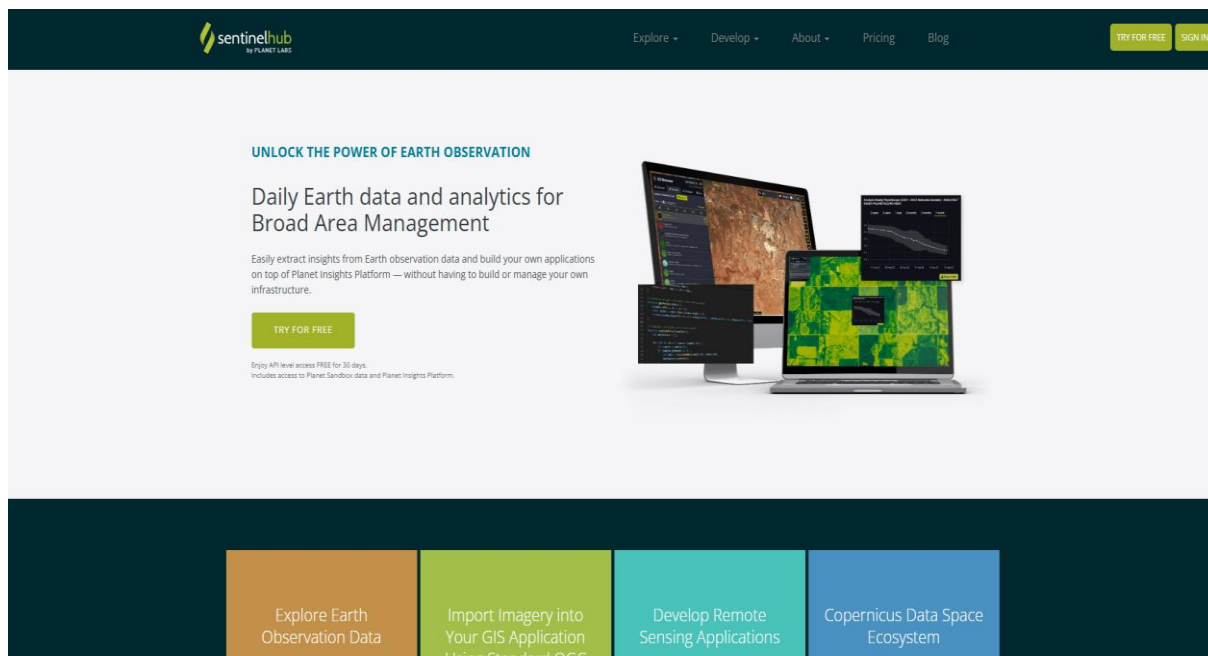


Рисунок 1.14 – Сервіс Sentinel Hub

Переваги:

- швидкий доступ до актуальних даних супутникового моніторингу;
- хмарна обробка дозволяє проводити аналіз без необхідності локальної потужної інфраструктури;
- інтуїтивно зрозумілий інтерфейс (наприклад, ео browser) для візуалізації результатів.

Недоліки:

- деякі можливості платформи доступні лише за підпискою;
- обмеження на гнучкість налаштування алгоритмів порівняно з власноруч розробленими рішеннями;
- може вимагатися додаткова інтеграція для реалізації складних задач сегментації.

1.4.3 Descartes Labs

Descartes Labs – це компанія, яка пропонує платформу геопросторової аналітики з використанням методів машинного навчання. Її сервіс дозволяє обробляти супутникові знімки, будувати моделі для класифікації ділянок та прогнозувати зміни в ландшафтах. Ключовим елементом підходу компанії є об'єднання великих обсягів даних із супутників, дронів та інших сенсорів в єдиний аналітичний простір, що забезпечує користувачам доступ до високоточних спостережень у масштабі всієї планети. Платформа також надає API для інтеграції з власними моделями глибинного навчання, а також можливості масштабованого обчислення у хмарному середовищі. Завдяки цьому Descartes Labs активно використовується у сферах агрономії, лісового господарства, енергетики та моніторингу кліматичних змін. Важливою особливістю є підтримка сценаріїв прогнозування – наприклад, аналіз сезонних змін рослинності або виявлення ознак деградації ґрунтів на основі історичних супутникових даних.

На рисунку 1.15 наведено сервіс Descartes Labs.

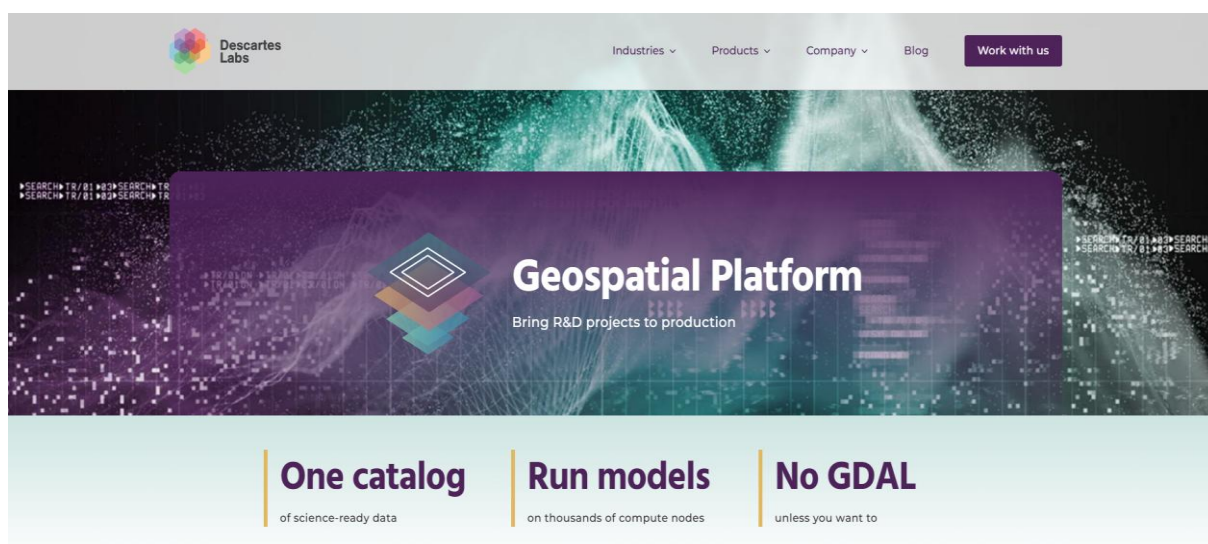


Рисунок 1.15 – Сервіс Descartes Labs

Переваги:

- використання передових алгоритмів глибокого навчання для аналізу супутникових даних;
- потужна інфраструктура обчислень, що забезпечує високу швидкодію обробки великих обсягів даних;
- гнучкий арі для інтеграції в бізнес-процеси та розробки кастомних рішень.

Недоліки:

- комерційна орієнтація сервісу може бути недоступною для малих дослідницьких груп через високу вартість;
- потреба в спеціалізованих знаннях для налаштування та використання платформи;
- можливе обмеження у доступі до деяких супутникових даних залежно від регіону.

1.4.4 Висновок щодо існуючих систем

На основі проведеного аналізу сервісів Google Earth Engine, Sentinel Hub та Descartes Labs можна зробити висновок, що сучасні глобальні платформи демонструють високий рівень автоматизації та ефективності в обробці супутникових знімків за допомогою методів штучного інтелекту.

Вони забезпечують доступ до великих обсягів даних, гнучкі API та потужні інструменти для аналізу, що дозволяють вирішувати широкий спектр завдань від моніторингу навколишнього середовища до планування міст.

Однак, на сьогодні в Україні відсутні подібні сервісні рішення, які б інтегрували всі ці можливості для автоматичної сегментації об'єктів та класифікації ділянок супутникових знімків.

Даний проект стане першим українським рішенням такого типу, що відкриває перспективи для розвитку дистанційного зондування та аналізу

геопросторових даних на національному рівні. Крім того, впровадження такого сервісу сприятиме підвищенню ефективності прийняття управлінських рішень у сферах екологічного та військового моніторингу, містобудування та сільського господарства, а також забезпечить конкурентоспроможність українських ІТ-технологій на світовому ринку.

2 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

2.1 Підготовка середовища перед експериментами

Для забезпечення надійних і відтворюваних результатів експериментальне середовище було ретельно спроектовано із використанням високопродуктивних обчислювальних ресурсів і стандартизованих фреймворків глибинного навчання. Ключові обчислювальні ресурси наведені у таблиці 2.1.

Таблиця 2.1 – Обчислювальні ресурси

| № | Категорія | Компоненти | Пояснення |
|---|--------------------------------------|-------------------------------|---|
| 1 | Апаратна конфігурація | GPU | NVIDIA RTX 3080 (10 ГБ відеопам'яті) |
| | | CPU | AMD Ryzen 9 7950X |
| | | RAM | 64 ГБ |
| | | Накопичувач | SSD на 1 ТБ |
| 2 | Програмне забезпечення та фреймворки | Python 3.8 | |
| | | TensorFlow 2.x та PyTorch 1.x | Для побудови та навчання нейронних мереж |
| | | OpenCV | Для обробки зображень |
| 3 | Характеристики датасету | Аугментація | Використано Albumentations для збільшення різноманіття навчальних даних |
| | | Роздільна здатність зображень | 512 × 512 та 1024 × 1024 пікселів |

Продовження таблиці 2.1

| | | | |
|--|--|------------------|--------------------------|
| | | Кількість класів | 3 (річки, ліси, будівлі) |
|--|--|------------------|--------------------------|

| | | | |
|--|--|-----------------|---|
| | | Розподіл набору | Навчання – 70%, валідація – 20%, тест – 10% |
|--|--|-----------------|---|

Завдяки ретельно створеному експериментальному середовищу було гарантовано оптимізацію результатів як за точністю, так і за обчислювальною ефективністю.

2.2 Навчання моделей та результати експериментів

Процес навчання супроводжувався тонким налаштуванням гіперпараметрів з метою досягнення максимальної точності сегментації. До процедури навчання було внесено такі зміни:

- було використано розміри пакета 8, 16 і 32 – для досягнення балансу між використанням пам'яті гри та швидкістю збіжності;
- початкову швидкість навчання було встановлено на рівні 0.001 зі ступеневим зменшенням до 0.0001 за методом `reducelronplateau`;
- було протестовано оптимізатори `adam` і `sgd`, причому `adam` забезпечив кращу стабільність на ранніх етапах навчання;
- кількість епох було встановлено на 100, при цьому застосовано механізм ранньої зупинки у випадку стабілізації валідаційних втрат;
- для оцінки втрат було застосовано `dice loss` (у випадку незбалансованих даних, зокрема для покращення сегментації дрібних об'єктів, таких як річки) та `categorical cross-entropy` (для багатокласової сегментації).

Вибір гіперпараметрів було здійснено за допомогою всебічного перебору сітки параметрів (`grid search`), що забезпечило досягнення найкращої продуктивності моделей. Навчені моделі було оцінено за стандартними метриками сегментації та порівняно між собою на основі різних архітектур. Зокрема, було проаналізовано показники IoU, Dice-коефіцієнта, піксельної точності та часу навчання на епоху для чотирьох

моделей – U-Net, DeepLabV3+, Mask R-CNN та Swin Transformer. Продуктивність кожної моделі наведено у таблиці 2.2.

Таблиця 2.2 – Ефективність моделей

| Модель | IoU (Перетин над об'єднанням) | Коефіцієнт Dice | Точність за пікселями | Час навчання (за одну епоху) |
|------------------|-------------------------------|-----------------|-----------------------|------------------------------|
| U-Net | 85.2% | 89.4% | 92.3% | 2 хв |
| DeepLabV3+ | 83.7% | 88.2% | 93.1% | 2 хв |
| Mask R-CNN | 81.5% | 86.8% | 91.5% | 3 хв |
| Swin Transformer | 86.8% | 90.1% | 92.8% | 4 хв |

На підставі отриманих результатів сформульовано наступні висновки. U-Net показала стабільні результати за всіма метриками, що дозволяє вважати її надійним варіантом для задач семантичної сегментації.

DeepLabV3+ досягнула найвищої піксельної точності, що є особливо важливим для сегментації великих площ, де критичною є узгодженість класифікації. Потужність цієї моделі забезпечується застосуванням атріозної згортки (atrous convolution) та багатомасштабної агрегації контексту.

Mask R-CNN надала можливість сегментації на рівні окремих об'єктів (instance-level), що є цінним для розрізнення кількох представників одного класу (наприклад, окремих будівель чи транспортних засобів). Водночас, спостерігалось дещо нижче значення IoU через складність обробки складного або зашумленого фону, характерного для природних ландшафтів. Це вказує на компроміс між точністю на рівні об'єктів та загальною семантичною узгодженістю.

Swin Transformer продемонстрував найкращі загальні показники за всіма метриками, що стало можливим завдяки ієрархічній архітектурі та

застосуванню механізмів самоуваги (self-attention), які ефективно моделюють просторові залежності на великих відстанях. Це дозволяє успішно виявляти тонкі просторові структури та контекст у супутникових зображеннях високої роздільності. Однак, така точність супроводжується високими обчислювальними витратами, що може обмежити практичне використання моделі в умовах обмежених ресурсів – наприклад, у реальному часі на борту супутника або на периферійних пристроях.

Попри досягнуті результати, було зафіксовано ряд проблемних аспектів:

- виникали помилки класифікації у прикордонних зонах (зокрема, вузькі річки іноді класифікувалися як дороги);
- спостерігалася варіативність умов освітлення та атмосфери (тіні та серпанок у супутникових зображеннях ускладнювали аналіз);
- виявлено нестачу даних для окремих регіонів (модель добре узагальнювала інформацію для типових ландшафтів, однак мала труднощі з менш поширеними середовищами).

2.3 Висновки щодо експериментів

Результати експериментів підтверджують ефективність моделей глибинного навчання для сегментації супутникових зображень, виявляючи суттєві відмінності в продуктивності залежно від архітектури. Високі значення показників IoU та коефіцієнта Dice свідчать про здатність моделей точно розрізняти різні типи земного покриття, зокрема річки, ліси та будівлі. Серед проаналізованих моделей Swin Transformer стабільно перевершувала традиційні архітектури на основі згорткових нейронних мереж (CNN), завдяки механізмам самоуваги, які ефективно вловлюють складні просторові залежності в супутникових зображеннях. U-Net, попри відносно просту архітектуру, показала конкурентоспроможні результати й залишається практичним вибором для задач масштабної сегментації через

свою обчислювальну ефективність та простоту навчання. DeepLabV3+ продемонструвала високу здатність до захоплення дрібних деталей, що особливо корисно при сегментації вузьких річок та невеликих об'єктів. Натомість Mask R-CNN виявилася придатною для сегментації на рівні окремих об'єктів (instance segmentation), однак стикалася з труднощами під час семантичної сегментації природних ландшафтів, головним чином через складність і варіативність текстур фону.

У ході аналізу було виявлено кілька ключових спостережень. Прикордонні зони між різними типами покриття викликали труднощі, часто спричиняючи помилки класифікації на межах. Дисбаланс даних також впливав на продуктивність моделей: області з меншою кількістю навчальних прикладів, наприклад, малопредставлені лісові зони, сегментувалися менш точно. Крім того, було встановлено, що здатність моделі до узагальнення значною мірою залежить від різноманітності датасету – моделі, навчені на географічно обмежених даних, часто мали труднощі із точною сегментацією ландшафтів у незнайомих регіонах. Ці висновки підкреслюють як сильні сторони, так і поточні обмеження методів сегментації на основі штучного інтелекту при їх застосуванні до реальних супутникових зображень.

Традиційні методи сегментації супутникових зображень, такі як порогова обробка, виявлення країв та класичні методи машинного навчання (наприклад, Random Forest або SVM), довгий час застосовувались у дистанційному зондуванні. Однак ці методи часто виявляються неефективними при роботі зі складними високодеталізованими зображеннями через обмежену здатність враховувати ієрархічні просторові зв'язки. Вони зазвичай базуються на ручному виділенні ознак та поверхневому представленні, що знижує їх ефективність у ситуаціях зі значними варіаціями текстур, освітлення та масштабу об'єктів. Як наслідок, їх точність суттєво знижується в умовах різноманітних ландшафтів або при роботі з великими та різноманітними супутниковими наборами даних.

Перелік переваг та недоліків моделей наведено у таблиці 2.3.

Таблиця 2.3 – Переваги та недоліки моделей

| Метод | Переваги | Недоліки |
|---|--|--|
| Порогова обробка та виявлення країв | Простота, швидкість, зрозумілість | Чутливість до умов освітлення та шуму |
| Random Forests і SVM | Ефективність для малих датасетів, інтерпретованість | Потребують ручного виділення ознак, обмежена масштабованість |
| Моделі на основі CNN (U-Net, DeepLabV3+) | Висока точність, здатність вивчати просторові ієрархії | Високі обчислювальні витрати |
| Моделі на основі трансформерів (Swin Transformer) | Виявлення довготривалих залежностей, найсучасніші результати | Потребують великих датасетів і значних обчислювальних ресурсів |

Результати свідчать про те, що методи глибинного навчання суттєво перевершують класичні підходи за точністю сегментації та стійкістю до шумів. Особливо архітектури на основі трансформерів демонструють високу ефективність при обробці складних супутникових зображень, що вказує на тенденцію переходу до таких моделей у сфері дистанційного зондування. Автоматична класифікація типів земного покриття за супутниковими знімками має численні прикладні застосування у різних сферах. У сфері моніторингу довкілля сегментація на основі ШІ дозволяє виявляти зміни у руслах річок, спричинені зміною клімату чи вирубкою лісів, що дає змогу дослідникам відстежувати деградацію природних ландшафтів з плином часу. Вона також сприяє оцінці зон потенційного підтоплення, що є важливою частиною стратегій запобігання стихійним лихам. Аналогічно, можливість аналізу втрати лісового покриття та

деградації ґрунтів допомагає природоохоронним організаціям та органам влади вжити відповідних заходів зі збереження довкілля. Міське планування та розвиток інфраструктури також значною мірою виграють від автоматизованих методів сегментації. Аналізуючи супутникові знімки, урбаністи можуть контролювати розширення міських територій, виявляти неформальні поселення та оцінювати зміни в землекористуванні. Такі дані є ключовими для розробки сталих міських стратегій і забезпечення ефективного розвитку інфраструктури. Автоматизована сегментація дає змогу органам влади відстежувати будівництво нових будівель і транспортних мереж, що сприяє прийняттю обґрунтованих рішень у великих інфраструктурних проєктах. Попри досягнення у сфері сегментації супутникових зображень на основі ШІ, все ще існують виклики, що обмежують широке впровадження таких рішень на практиці. Однією з основних проблем є узагальнення моделей для різних географічних регіонів. Супутникові знімки значно варіюються залежно від атмосферних умов, типів рослинності та міської забудови, що ускладнює застосування моделей, навчених на одному наборі даних, до інших регіонів. Це обмеження вимагає застосування методів адаптації до нових доменів або збирання більш різноманітних навчальних даних для підвищення стійкості моделей. Іншою суттєвою проблемою є дисбаланс класів і виявлення рідкісних об'єктів. У багатьох супутникових наборах даних деякі типи земного покриву, такі як річки або будівлі, зустрічаються значно рідше, ніж домінуючі класи – наприклад, ліси чи відкриті простори. Такий дисбаланс призводить до упереджених прогнозів моделей, у яких рідкісні класи часто неправильно класифікуються або ігноруються. Для вирішення цієї проблеми необхідно застосовувати спеціалізовані підходи, зокрема аугментацію даних, функцію втрат *focal loss* і генерацію синтетичних даних, щоб забезпечити збалансоване навчання.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір мови програмування та технологій

У рамках реалізації веб-додатку для семантичної сегментації супутникових зображень ключовим завданням було визначити відповідний стек технологій, що забезпечував би стабільну та ефективну роботу як клієнтської, так і серверної частин системи. Для бекенду було обрано мову програмування TypeScript у середовищі Node.js. Це рішення пояснюється тим, що TypeScript, як надмножина JavaScript, забезпечує статичну типізацію та кращу читаємість коду, що особливо важливо при роботі з обробкою даних та взаємодією з нейронною мережею. Node.js, у свою чергу, надає можливість реалізувати високопродуктивний сервер на основі асинхронної подієвої моделі, що є доречним для обробки HTTP-запитів на обробку зображень.

Для реалізації машинного навчання використовується попередньо навчена модель сегментації типу U-Net, яка експортується у форматі ONNX. Цей формат дозволяє забезпечити сумісність із різними фреймворками і дає змогу використовувати модель безпосередньо на сервері завдяки бібліотеці onnxruntime-node. Для обробки зображень, отриманих із супутника або карти, додатково використовується бібліотека jimp, яка дозволяє змінювати розмір, зчитувати або перетворювати зображення у відповідні формати.

Фронтенд частина побудована з використанням бібліотеки React, яка дозволяє створювати зручний та динамічний інтерфейс користувача. Для стилізації застосовано Tailwind CSS – фреймворк з набором готових класів для побудови адаптивного дизайну без написання окремих CSS-файлів. Обмін даними між клієнтом та сервером здійснюється за допомогою бібліотеки Axios. Візуалізація результатів сегментації у вигляді кругової діаграми реалізована за допомогою Recharts – бібліотеки, яка легко інтегрується з React.

Для отримання супутникових знімків використовується Google Maps Static API, а також Selenium WebDriver, який дозволяє у напівавтоматичному режимі знімати карту з географічними координатами, вказаними користувачем. Для безпечної роботи з конфіденційною інформацією, зокрема API-ключами, було впроваджено використання змінних середовища з бібліотекою dotenv.

Таким чином, обрані технології дозволили ефективно реалізувати як логіку обробки зображень та запуску моделі, так і інтерактивну частину взаємодії з користувачем у межах єдиного веб-додатку. Такий підхід забезпечує гнучкість системи, зручність у розгортанні та потенціал до масштабування або подальшого розвитку проєкту.

3.2 Вибір та підготовка датасету

Для навчання моделі сегментації було необхідно використати репрезентативний датасет супутникових зображень з наявною піксельною розміткою, яка дозволяє визначати клас кожного окремого пікселя на зображенні. У проєкті було обрано відкритий датасет Poland Aerial Dataset – набір високоякісних аерофотознімків території Польщі з деталізованими масками, що відображають різні типи ландшафту: забудови, зелені насадження, водні об'єкти, дороги та фонові ділянки. Такий вибір пояснюється як доступністю даних, так і їхньою придатністю для вирішення завдання семантичної сегментації на рівні урбанізованих територій.

Кожне зображення в датасеті має розмір 256×256 пікселів та супроводжується відповідною маскою, де кожен колірний канал відповідає певному класу об'єкта. Для зручності використання ці зображення були попередньо перетворені у формат, зручний для моделі U-Net, а саме: обрані найбільш збалансовані за розподілом класів знімки, приведені до однакового розміру, нормалізовані значення пікселів та збережені у форматі, придатному для експорту моделі у формат ONNX.

Попереднє навчання моделі здійснювалося поза межами основного веб-додатку – в окремому середовищі, яке не включено до репозиторію. Після завершення навчання модель експортується у формат .onnx, що дозволяє її запуск без використання важких фреймворків (наприклад, PyTorch) безпосередньо на сервері, що обробляє запити користувача.

Даний датасет також був обраний через схожість з українськими територіями за своєю структурою, типом забудови та характером ландшафту, що є досить близьким до умов центральної та західної України. Це пояснюється географічною близькістю, кліматичними подібностями та подібними урбаністичними особливостями регіонів. Тому, даний набір даних є технічно придатним для моделювання сегментації українських територій, та використання у даному веб-додатку.

Таким чином, вибір саме Poland Aerial Dataset зумовлений його якістю, тематичною відповідністю та сумісністю з обраною архітектурою нейронної мережі. Його структура дозволила забезпечити точність сегментації та коректну роботу моделі в режимі inference при інтеграції у веб-сервіс.

3.3 Архітектура веб-додатку

Розроблений веб-додаток побудовано за класичною архітектурою клієнт–сервер, де фронтенд (клієнтська частина) відповідає за взаємодію з користувачем, а бекенд (серверна частина) – за обробку запитів, завантаження зображень та виконання інференсу моделі сегментації. Така архітектура є гнучкою, масштабованою та дозволяє розділяти відповідальність між окремими компонентами, що спрощує супровід системи та можливе розширення її функціоналу в майбутньому.

Користувач взаємодіє з інтерфейсом через веб-браузер, у якому представлено форму для введення географічних координат. Після натискання кнопки запуску запиту, клієнтська частина надсилає HTTP-

запит до бекенду із зазначеними широтою та довготою. Серверна частина приймає ці координати та за допомогою Google Maps API, та отримує супутникове зображення відповідної ділянки. Це зображення зберігається локально та передається на вхід нейронній мережі. Результатом роботи моделі є карта піксельної класифікації, яка в подальшому використовується для побудови сегментованого зображення.

Обидва зображення – оригінальне та сегментоване – перетворюються у формат base64 та відправляються у відповідь клієнту разом із статистикою по класах (кількість пікселів кожного типу). Фронтенд приймає ці дані та виводить їх у зручному вигляді: два зображення (до та після обробки) і кругова діаграма для візуалізації розподілу об'єктів по класах. Також реалізовано легенду, яка пояснює відповідність кольорів сегментації конкретним типам об'єктів на зображенні.

Таким чином, архітектура веб-додатку є чітко поділеною на логічні блоки:

а) користувач (Client):

- вводить координати (широта, довгота);
- надсилає запит через інтерфейс (react);

б) фронтенд (React App):

- відображає карту;
- приймає координати;
- надсилає запит на бекенд;
- отримує: оригінальне зображення, сегментоване зображення, статистику класів;
- відображає: результат (зображення, діаграма та панель «Детальніше»);

в) бекенд (Node.js + TypeScript):

- приймає запит із координатами;
- викликає модуль завантаження зображення: google maps static api;

- зберігає зображення (queried_image.png);
- передає в модуль інференсу onnx-моделі;

г) модуль ONNX-інференсу:

- завантажує модель unet_poland_ds_modelv1.onnx;
- преобразує зображення в тензор;
- виконує сегментацію;
- повертає маску сегментації (матрицю класів);

д) генератор зображення (Canvas):

- перетворює матрицю в png;
- розфарбовує пікселі відповідно до класів;
- рахує кількість пікселів кожного класу;

е) відповідь сервера Фронтенду JSON:

- ori_img: <base64>;
- pred_result: <base64>;
- class_count: [int, int, int, int, int].

Архітектура веб-додатку наведена на рисунку 3.1.

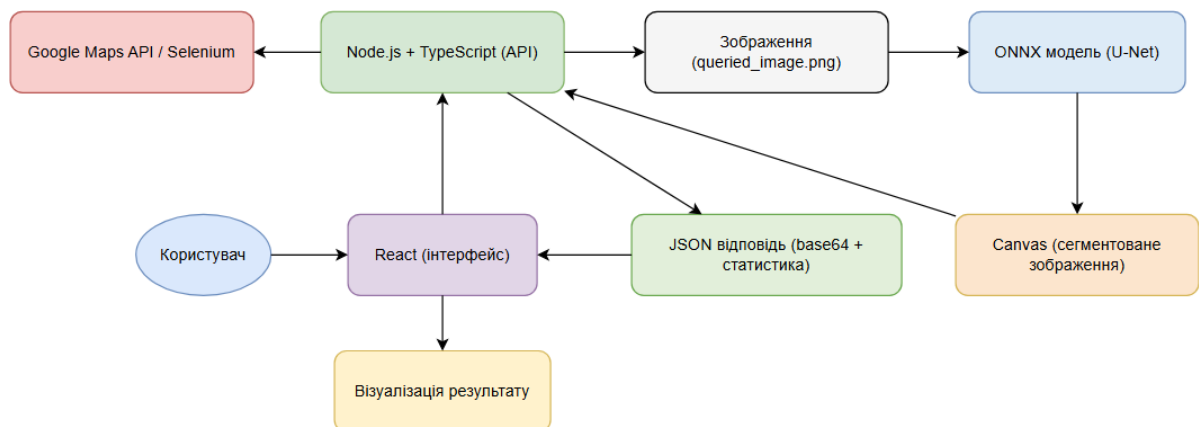


Рисунок 3.1 – Архітектура веб-додатку

Дана схема дозволяє не лише забезпечити зручну роботу користувача, але й легко адаптувати систему для інших типів моделей або картографічних джерел у майбутньому.

3.4 Розробка серверної частини (Backend)

Серверна частина веб-додатку виконує критичну роль у загальній архітектурі системи, оскільки забезпечує взаємодію між користувачем, зовнішніми картографічними API та моделлю глибокого навчання. Для її реалізації було обрано середовище виконання Node.js з мовою програмування TypeScript, що поєднує переваги гнучкого JavaScript і можливість типізації, характерну для строго типізованих мов.

Основним елементом бекенду є HTTP-сервер, що «прослуховує» запити на маршруті `/api/segment_query`. На вхід він приймає географічні координати (широту та довготу), які передаються через параметри запиту. Після отримання координат сервер виконує завантаження супутникового зображення з використанням Google Maps Static API – інструменту автоматизації браузера, що дозволяє отримати зображення з інтерактивної карти при повному контролі над інтерфейсом.

Завантажене зображення зберігається у вигляді PNG-файлу й надалі використовується як вхід до нейронної мережі. Обробка зображення здійснюється за допомогою бібліотеки `ijmp`, яка дозволяє масштабувати, зчитувати та отримувати доступ до піксельних значень. Після цього зображення перетворюється у тензор формату $[1, 256, 256, 3]$, що відповідає вимогам моделі.

Для запуску сегментації використовується бібліотека `onnxruntime-node`, що дозволяє виконувати інференс моделі у форматі ONNX без використання важких фреймворків. Результатом є двовимірний масив, у якому кожному пікселю відповідає індекс найбільш імовірного класу. Отримана карта класифікації візуалізується на новому полотні з використанням бібліотеки `canvas`, а кольори пікселів призначаються відповідно до легенди, визначеної в додатку.

Крім графічного результату, сервер також обчислює кількість пікселів, що відповідають кожному з класів (фон, будівлі, зелень, вода,

дороги), і формує відповідну статистику. Після завершення обробки сервер кодує як оригінальне, так і сегментоване зображення у формат base64, додає масив кількостей пікселів по класах та відправляє JSON-відповідь на клієнт.

Для забезпечення безпеки та конфіденційності доступу до API-сервісів у проєкті використовується файл .env, де зберігаються такі змінні середовища, як ключ до Google Maps API. Це дозволяє відокремити конфігурацію від вихідного коду та підвищити безпеку при розгортанні проєкту на сервері або в хмарі.

3.5 Розробка клієнтської частини (Frontend)

Розробка клієнтської частини веб-додатку мала на меті створити інтуїтивно зрозумілий, зручний та візуально привабливий інтерфейс, який дозволяє користувачеві легко взаємодіяти з системою сегментації супутникових зображень. Основним інструментом для реалізації фронтенду було обрано React – сучасну JavaScript-бібліотеку для створення динамічних односторінкових додатків (SPA). Вона дозволяє ефективно управляти станом компонента, обробляти запити до сервера та оновлювати вміст інтерфейсу без перезавантаження сторінки.

Форма на інтерфейсі містить два основні поля – широта та довгота. Після введення координат користувач натискає кнопку запуску, що активує функцію onSubmit, яка надсилає HTTP-запит на бекенд за допомогою бібліотеки Axios. У відповідь від сервера клієнт отримує:

- оригінальне зображення у форматі base64;
- сегментоване зображення після проходження через модель;
- та статистичні дані щодо кількості пікселів кожного класу.

Ці результати виводяться на екран у вигляді двох зображень (до та після обробки), а також кругової діаграми, яка візуалізує розподіл класів. Для створення цієї діаграми використано бібліотеку Recharts, яка легко інтегрується з React і підтримує адаптивну побудову графіків. Усі візуальні

елементи інтерфейсу стилізовані за допомогою Tailwind CSS – CSS-фреймворку з класовою утилітарною структурою, що забезпечує швидку та гнучку верстку без необхідності писати окремі стилі вручну.

Також було реалізовано компонент легенди, який наочно демонструє відповідність між кольорами сегментації та конкретними типами об'єктів (наприклад, будівлі, вода, дорога). Реалізацію веб-сторінки представлено на рисунку 3.2.



Рисунок 3.2 – Головна сторінка веб-додатку

3.6 Реалізація моделі сегментації

У центрі розробленого веб-додатку лежить модель семантичної сегментації, яка виконує основну аналітичну функцію – класифікацію кожного пікселя супутникового зображення до одного з п'яти заздалегідь визначених класів: фон, будівлі, зелена зона, вода, дорога. Для цього використано архітектуру U-Net, яка зарекомендувала себе як ефективне рішення в задачах піксельної сегментації, зокрема у медичних, урбаністичних та геопросторових даних.

Навчання моделі здійснювалося на окремому етапі, поза межами веб-додатку, із використанням Python та PyTorch. Після навчання модель було

експортовано у формат ONNX (Open Neural Network Exchange), який забезпечує платформи-незалежне представлення моделі та дозволяє запускати її у середовищах, де відсутній сам фреймворк.

У веб-додатку inference, тобто прогнозування, здійснюється на сервері за допомогою бібліотеки `onnxruntime-node`, що надає інтерфейс для завантаження моделі, перетворення вхідних даних у тензори та обробки їх безпосередньо у середовищі Node.js. Перед подачею зображення на вхід воно перетворюється у тензор формату [1, 256, 256, 3]:

- 1 – кількість зображень у пакеті (`batch size = 1`, тобто модель обробляє одне зображення за раз);
- 256 – висота зображення в пікселях;
- 256 – ширина зображення в пікселях;
- 3 – кількість каналів (RGB: червоний, зелений, синій), тобто кольорове зображення.

Що відповідає структурі очікуваних даних моделі. Значення пікселів нормалізуються до діапазону [0, 1], і лише після цього запускається інференс.

На виході модель генерує багатовимірний масив, у якому для кожного пікселя зображення подається набір ймовірностей належності до кожного з п'яти класів. У веб-додатку реалізовано постобробку: для кожного пікселя обирається клас із найвищою ймовірністю (`argmax`), і на основі цього формується масив піксельних індексів. Цей масив потім візуалізується у вигляді кольорового зображення, де кожному класу відповідає визначений колір.

Уся ця реалізація дозволяє запускати повноцінну нейронну модель безпосередньо у JavaScript-середовищі на сервері, не використовуючи повноцінне середовище Python. Це значно спрощує розгортання проекту, дозволяє обійтися без складного оточення і полегшує інтеграцію з фронтендом. Крім того, швидкість обробки одного зображення є

прийнятною для інтерактивного веб-сервісу – обробка займає до кількох секунд залежно від потужності хосту.

Таким чином, модель U-Net у форматі ONNX разом із середовищем `onnxruntime-node` стала центральним елементом обчислювальної логіки веб-додатку, забезпечуючи точну, відтворювану й масштабовану сегментацію супутникових зображень.

3.7 Інтеграція моделі в веб-додаток

Інтеграція моделі сегментації в веб-додаток є ключовим етапом, який поєднує аналітичну частину (інференс нейронної мережі) з користувацьким інтерфейсом. Головна мета цього етапу – зробити так, щоб результат роботи моделі був безпосередньо доступний користувачу у зручному та наочному форматі без необхідності ручного запуску чи обробки.

Модель сегментації, попередньо навчена та експортована у форматі `.onnx`, зберігається на сервері в папці `model/` і завантажується під час кожного запиту користувача. Її обробка реалізована в окремій асинхронній функції, яка виконує наступні дії: зчитування супутникового зображення, перетворення його у тензор, запуск інференсу, формування карти класифікації пікселів та побудова сегментованого зображення. Усі ці кроки реалізовані в рамках одного HTTP-запиту до маршруту `/api/segment_query`, що приймає широту та довготу як параметри.

Код функціоналу побудований таким чином, що інференс моделі, обробка зображення та генерація відповіді відбуваються в одному логічному потоці. Після завершення обробки результати – а саме оригінальне зображення, сегментоване зображення та масив кількостей пікселів кожного класу – кодуються у формат `base64` та передаються клієнтській частині як частина JSON-відповіді. Завдяки цьому інтеграція з фронтендом є прозорою і не вимагає додаткових звернень до файлової системи чи окремих API.

З боку фронтенду React-компонент отримує дані, розпізнає base64-рядки як зображення і виводить їх на екран у відповідних блоках. Статистика піксельного розподілу по класах передається до компоненту кругової діаграми, що відображає результати візуально. Таким чином, користувач у режимі реального часу отримує інтерактивну відповідь на свій запит: спочатку – супутникове зображення за заданими координатами, а потім – результат автоматичної семантичної сегментації.

3.8 Візуалізація результатів

У реалізованій системі візуалізація результатів відбувається на двох рівнях: виведення оброблених зображень та графічне представлення статистики сегментації. На інтерфейсі користувача відображаються дві основні візуальні панелі. Перша містить оригінальне супутникове зображення, отримане за введеними координатами. Друга – сегментоване зображення, згенероване в результаті інференсу моделі, у якому кожному пікселю присвоєно колір відповідно до класу, до якого він належить. Приклад наведено на рисунку 3.3.

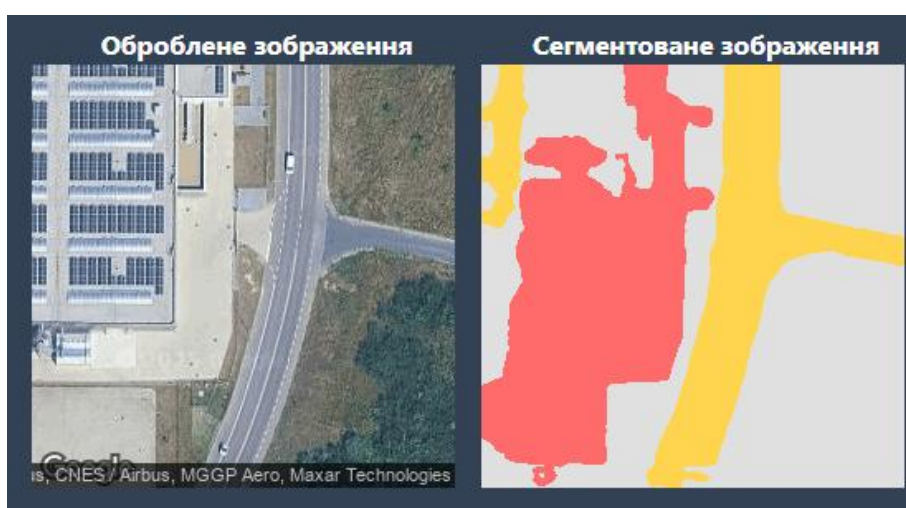


Рисунок 3.3 – Оригінальне супутникове зображення та сегментоване зображення

Для розфарбовування пікселів після обробки зображення було обрано наступну палітру:

- сірий (фон);
- рожевий (будівлі);
- зелений (рослинність);
- блакитний (вода);
- жовтий (дороги).

Ці кольори відображаються як на самому сегментованому зображенні, так і у панелі «Детальніше», що розміщується поруч. Панель «Детальніше» реалізована у вигляді вертикального списку з кольоровими мітками та назвами класів. Такий підхід дозволяє інтуїтивно співвіднести зображення на екрані з реальними об'єктами місцевості.

Приклад наведено на рисунку 3.4.

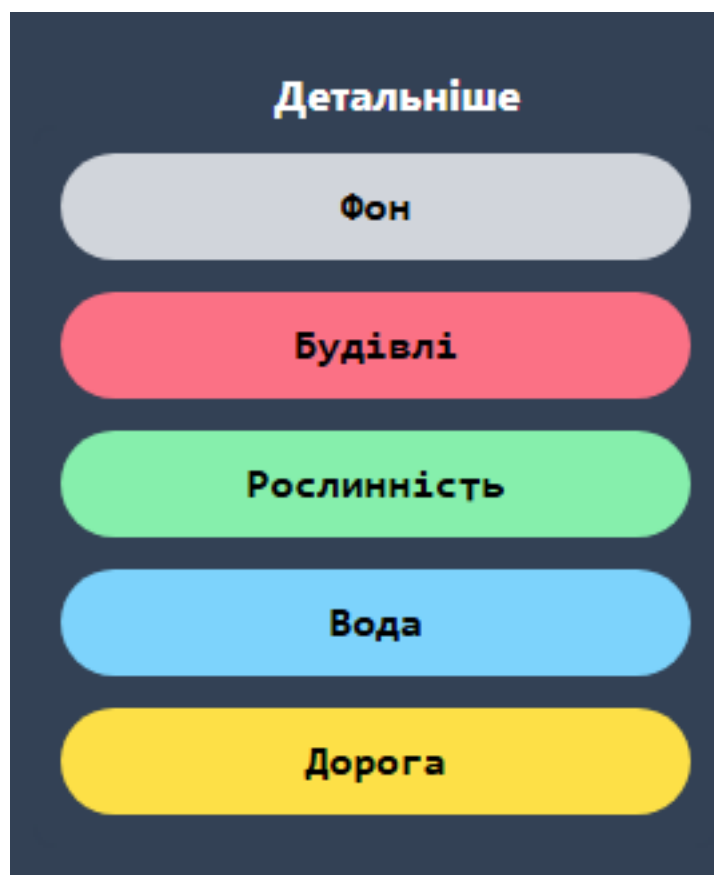


Рисунок 3.4 – Панель «Детальніше»

Для кількісного аналізу результатів сегментації використовується кругова діаграма, реалізована за допомогою бібліотеки Recharts, що інтегрована у React. Ця діаграма будується на основі масиву, який повертає сервер: кожному з п'яти класів відповідає кількість пікселів, що належать до нього. Усі сегменти діаграми мають ті самі кольори, що й у зображенні, завдяки чому забезпечується візуальна узгодженість інтерфейсу. Приклад наведено на рисунку 3.5.



Рисунок 3.5 – Кругова діаграма результату сегментації

3.9 Тестування системи

Після завершення основних етапів реалізації веб-додатку було проведено тестування з метою перевірки точності роботи та швидкодії моделі сегментації. У середньому повний цикл (запит → отримання зображення → сегментація → візуалізація) займав від 2 до 3 секунд, що є цілком прийнятним результатом. Приклади тестування точності роботи моделі наведені на рисунках 3.6–3.10.

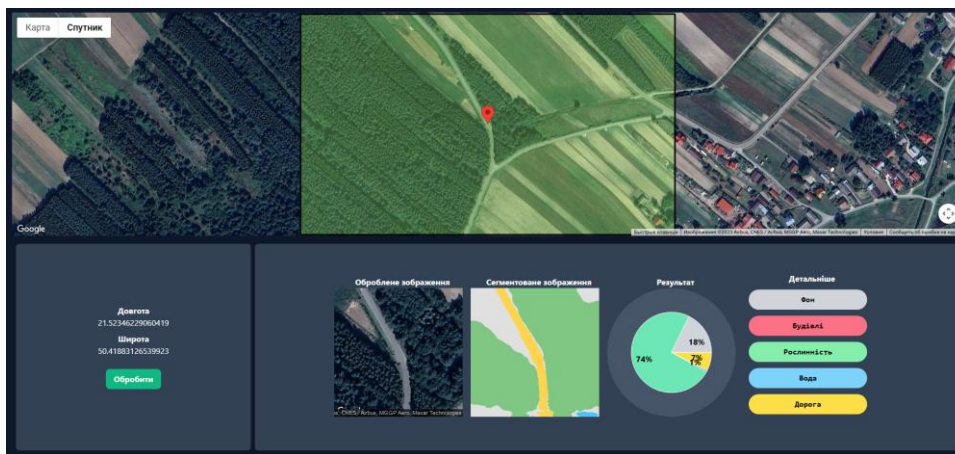


Рисунок 3.6 – Тестування точності моделі №1

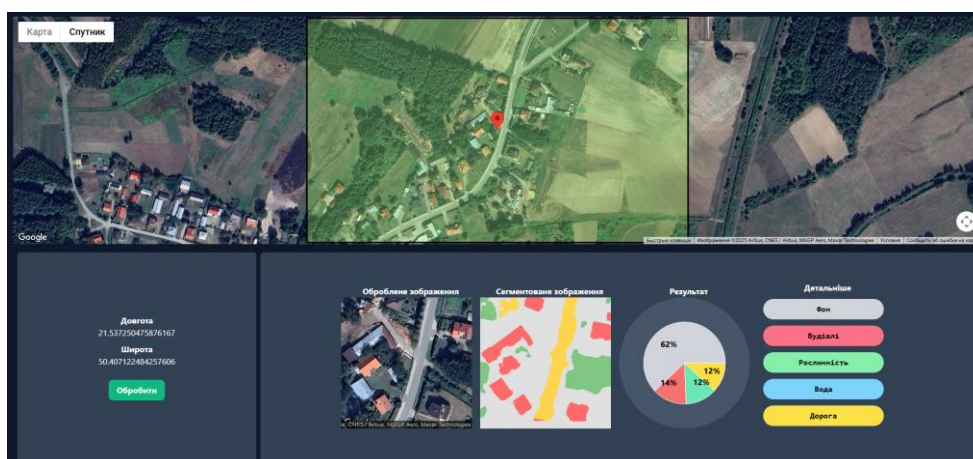


Рисунок 3.7 – Тестування точності моделі №2

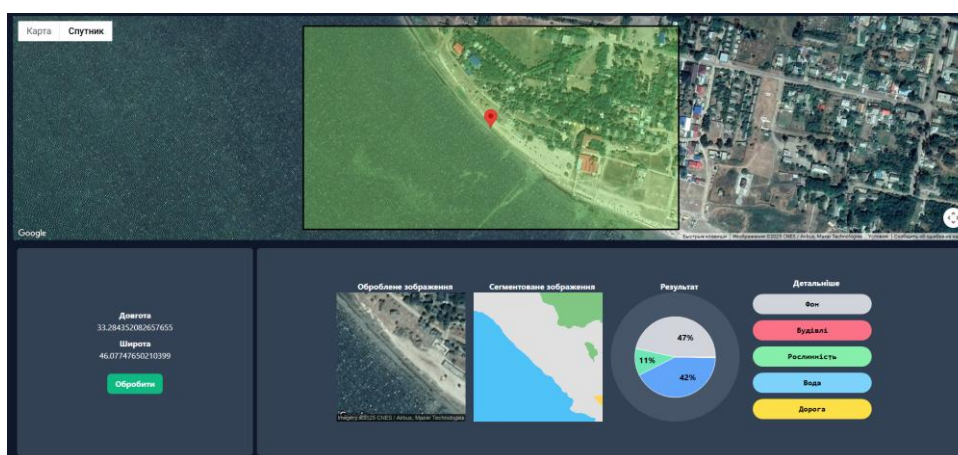


Рисунок 3.8 – Тестування точності моделі №3

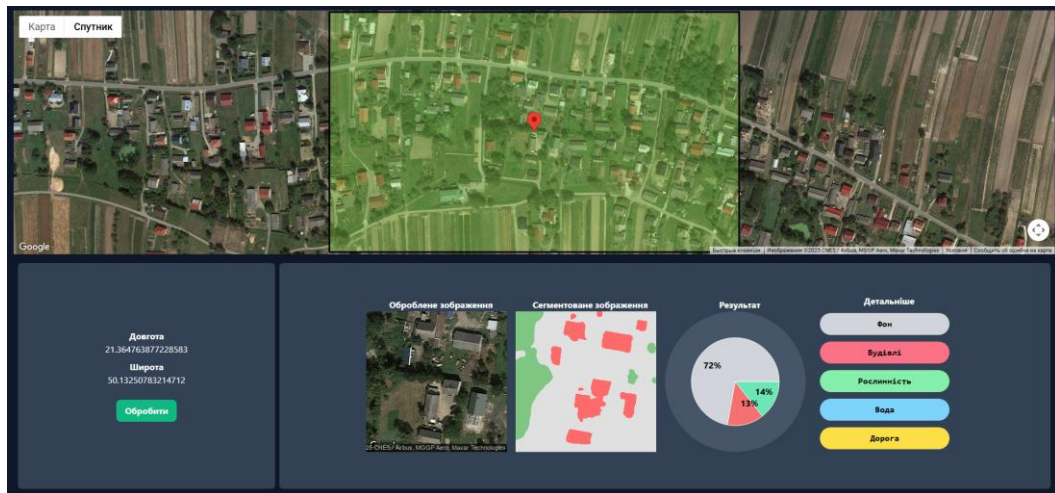


Рисунок 3.9 – Тестування точності моделі №4

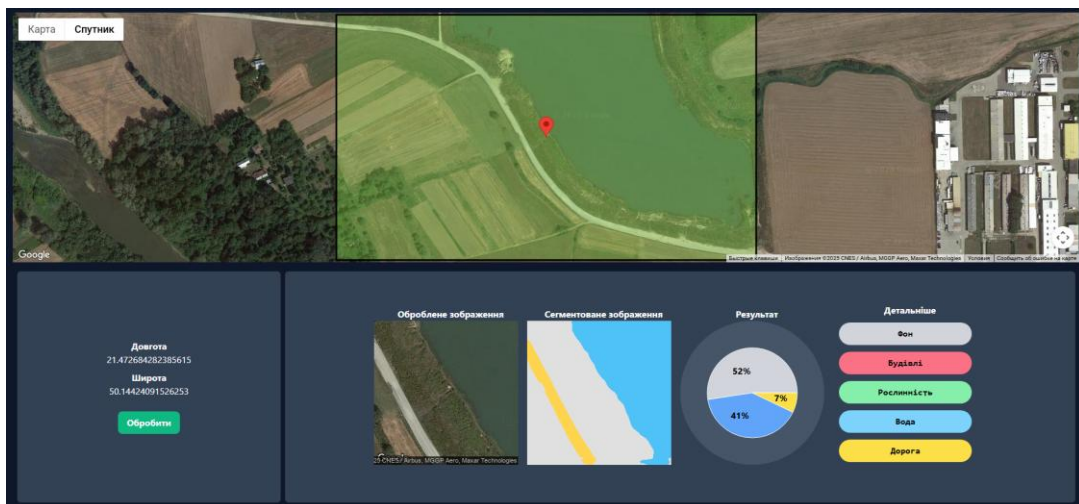


Рисунок 3.10 – Тестування точності моделі №5

У підсумку можна стверджувати, що модель виконує поставлені функції не ідеально, але з високою точністю та стабільністю, а проведене тестування підтвердило його придатність до використання.

3.10 Висновки щодо розробленої системи

У результаті розробки вдалося створити повноцінну клієнт-серверну систему, яка дозволяє користувачу за координатами місцевості отримувати

автоматично оброблені супутникові знімки з відокремленням основних класів об'єктів – таких як фон, будівлі, рослинність, вода та дороги.

Побудова системи включала розробку користувацького інтерфейсу, серверної логіки, інфраструктури для отримання супутникових зображень, а також інтеграцію попередньо навченої моделі сегментації у форматі ONNX.

Однією з головних переваг розробленого підходу є його модульність і гнучкість. Система легко адаптується до інших типів моделей, змін джерела супутникових зображень, а також до можливості розгортання у хмарному середовищі. У межах тестування було підтверджено стабільність роботи додатку, прийнятну швидкість та точність моделі на цільових ділянках місцевості.

Водночас у ході роботи було виявлено низку напрямів для подальшого розвитку. До них можна віднести:

- розширення класів сегментації за рахунок донавчання моделі на локальних даних;
- перехід до більш високої роздільної здатності зображень;
- додавання режиму порівняння змін на місцевості (наприклад, сегментація до і після);
- реалізацію особистого кабінету з історією запитів;
- оптимізацію продуктивності;
- додавання підтримки альтернативних джерел знімків, зокрема супутників Copernicus або Planet.

ВИСНОВКИ

У результаті проведеного дослідження було досягнуто мети кваліфікаційної роботи – дослідження, порівняння та реалізація методів семантичної сегментації супутникових зображень із використанням алгоритмів штучного інтелекту.

Проведено аналіз сучасних підходів до сегментації, зокрема архітектур U-Net, DeepLabV3+, Mask R-CNN та Swin Transformer. На основі результату експериментів встановлено, що моделі глибинного навчання значно перевершують традиційні методи за точністю, стійкістю до шуму та здатністю до генералізації.

Розроблена клієнт-серверна система демонструє можливість інтеграції моделі сегментації у веб-додаток із повною автоматизацією процесу: від отримання супутникового зображення до виводу результату з візуалізацією.

Модель U-Net, імплементована у форматі ONNX, забезпечила прийнятну точність та швидкодію, а веб-інтерфейс, створений на базі React та Tailwind, забезпечив доступність і зручність для користувача.

Система була протестована на реальних зображеннях, що підтвердило її ефективність. Основні переваги розробки – модульність, гнучкість, можливість масштабування та адаптації до нових датасетів.

Подальші напрямки роботи включають:

- донавчання моделі на локальних українських даних;
- підтримку вищої роздільної здатності знімків;
- реалізацію аналізу змін місцевості у динаміці;
- підключення альтернативних картографічних API;
- створення особистого кабінету користувача.

Робота пройшла апробацію у межах «Восьмого міжнародного семінара з комп'ютерного моделювання та інтелектуальних систем (CMIS-2025)», 5 травня 2025 року, м. Запоріжжя.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. AI Models for Automatic Objects Classification in Satellite Images / V. Vysotska, K. Smelyakov, S. Osiievskyi and V. Yartsev. CMIS-2025: Eighth International Workshop on Computer Modeling and Intelligent Systems, May 5, 2025, Zaporizhzhia, Ukraine.
2. Artificial intelligence techniques for landslides prediction using satellite imagery / A. Sharma. IEEE access. 2024. C. 1–18. URL: <https://doi.org/10.1109/access.2024.3446037> (date of access: 02.05.2025).
3. A dual branch multi-scale stereo matching network for high-resolution satellite remote sensing images / Z. Xu. IEEE journal of selected topics in applied earth observations and remote sensing. 2024. C. 1–17. URL: <https://doi.org/10.1109/jstars.2024.3502842> (date of access: 02.05.2025).
4. An epipolar resampling method for multi-view high resolution satellite images based on block / H. Yi. IEEE access. 2021. T. 9. C. 162884–162892. URL: <https://doi.org/10.1109/access.2021.3133664> (date of access: 03.05.2025).
5. Cho E., Kim E., Choi Y. Cloud cover prediction model using multi-channel geostationary satellite images. IEEE transactions on geoscience and remote sensing. 2024. C. 1–15. URL: <https://doi.org/10.1109/tgrs.2024.3473992> (date of access: 04.05.2025).
6. Deep convolutional network based machine intelligence model for satellite cloud image classification / K. K. Jena та ін. Big data mining and analytics. 2023. T. 6, № 1. C. 1–12. URL: <https://doi.org/10.26599/bdma.2021.9020017> (date of access: 04.05.2025).
7. GAN-based satellite imaging: a survey on techniques and applications / H. Mansourifar та ін. IEEE access. 2022. C. 1. URL: <https://doi.org/10.1109/access.2022.3221123> (date of access: 06.05.2025).
8. Hu Z., Zhang K., Liu Y. Edge constrained DSM refinement based on shading from high resolution multi-view satellite images. IEEE journal of selected

topics in applied earth observations and remote sensing. 2025. C. 1–12. URL: <https://doi.org/10.1109/jstars.2025.3526817> (date of access: 06.05.2025).

9. Illegal microdumps detection in multi-mission satellite images with deep neural network and transfer learning approach / C. Marrocco та ін. IEEE access. 2024. C. 1. URL: <https://doi.org/10.1109/access.2024.3409393> (date of access: 06.05.2025).

10. Karwowska K., Wierzbicki D. Using super-resolution algorithms for small satellite imagery: a systematic review. IEEE journal of selected topics in applied earth observations and remote sensing. 2022. C. 1. URL: <https://doi.org/10.1109/jstars.2022.3167646> (date of access: 07.05.2025).

11. Land-Cover classification with high-resolution remote sensing images using interactive segmentation / L. Xu та ін. IEEE access. 2022. C. 1. URL: <https://doi.org/10.1109/access.2022.3205327> (date of access: 07.05.2025).

12. Methods for the epipolarity analysis of pushbroom satellite images based on the rational function model / H. Yi та ін. IEEE access. 2020. T. 8. C. 103973–103983. URL: <https://doi.org/10.1109/access.2020.2999393> (date of access: 07.05.2025).

13. Ouchra H., Belangour A., Erraissi A. Machine learning algorithms for satellite image classification using Google Earth Engine and Landsat satellite data: morocco case study. IEEE access. 2023. C. 1. URL: <https://doi.org/10.1109/access.2023.3293828> (date of access: 07.05.2025).

14. Sasaki K., Sekine T., Emery W. Enhancing the detection of coastal marine debris in very high resolution satellite imagery via unsupervised domain adaptation. IEEE journal of selected topics in applied earth observations and remote sensing. 2024. C. 1–16. URL: <https://doi.org/10.1109/jstars.2024.3364165> (date of access: 08.05.2025).

15. Segmentation mask refinement using image transformations / T. D. Nguyen та ін. IEEE access. 2017. T. 5. C. 26409–26418. URL: <https://doi.org/10.1109/access.2017.2772269> (date of access: 08.05.2025).

16. Shakya S., Kumar S., Goswami M. Deep learning algorithm for satellite imaging based cyclone detection. IEEE journal of selected topics in applied earth observations and remote sensing. 2020. T. 13. C. 827–839. URL: <https://doi.org/10.1109/jstars.2020.2970253> (date of access: 08.05.2025).

17. Shende S. CNN based missing object detection. International journal for research in applied science and engineering technology. 2023. T. 11, № 4. C. 956–959. URL: <https://doi.org/10.22214/ijraset.2023.50138> (date of access: 08.05.2025).

18. Surface depth estimation from multi-view stereo satellite images with distribution contrast network / Z. Chen та ін. IEEE journal of selected topics in applied earth observations and remote sensing. 2024. C. 1–10. URL: <https://doi.org/10.1109/jstars.2024.3457616> (date of access: 09.05.2025).

19. TCCL-DenseFuse: infrared and water vapor satellite image fusion model using deep learning / C.-J. Zhang та ін. IEEE journal of selected topics in applied earth observations and remote sensing. 2023. C. 1–25. URL: <https://doi.org/10.1109/jstars.2023.3277842> (date of access: 09.05.2025).

20. Vegetation land use/land cover extraction from high-resolution satellite images based on adaptive context inference / Z. Zhan та ін. IEEE access. 2020. T. 8. C. 21036–21051. URL: <https://doi.org/10.1109/access.2020.2969812> (date of access: 09.05.2025).

21. YOLO-OSD: optimized ship detection and localization in multi-resolution SAR satellite images using a hybrid data-model centric approach / M. F. Humayun та ін. IEEE journal of selected topics in applied earth observations and remote sensing. 2024. C. 1–20. URL: <https://doi.org/10.1109/jstars.2024.3365807> (date of access: 09.05.2025).

22. K. Smelyakov, S. Smelyakov and A. Chupryna, Advances in Spatio-Temporal Segmentation of Visual Data, in Adaptive Edge Detection Models and Algorithms. – Springer Nature Switzerland AG 2020, pp. 1–51. URL: https://doi:10.1007/978-3-030-35480-0_1 (date of access: 10.05.2025).

23. S. Voloshyn, та ін., «Big Data Analysis for Multispectral Images Recognition Based on Deep Learning», IEEE 16th International Conference on Computer Sciences and Information Technologies, vol. 1, pp. 160-170, 2021. URL: <https://doi:10.1109/CSIT52700.2021.9648650> (date of access: 10.05.2025).
24. A. Sartiukova, та ін., «The Multiclass Classification of Objects Based on Multispectral Images Recognition», IEEE 16th International Conference on Computer Sciences and Information Technologies, vol. 1, pp. 52-60, 2021. URL: <https://doi:10.1109/CSIT52700.2021.9648719> (date of access: 10.05.2025).
25. K. Smelyakov, P. Dmitry, M. Vitalii and C. Anastasiya, «Investigation of network infrastructure control parameters for effective intellectual analysis», 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2018, pp. 983-986, URL: <https://doi:10.1109/TCSET.2018.8336359> (date of access: 10.05.2025).
26. S. Tchynetskyi, та ін., «A Neural Network Development for Multispectral Images Recognition», IEEE 16th International Conference on Computer Sciences and Information Technologies, vol. 2, pp. 278-284, 2021. URL: <https://doi:10.1109/CSIT52700.2021.9648735> (date of access: 10.05.2025).
27. Li H., Yuan G.-L., Xu C. Siamese contour segmentation network for multi-state object tracking. SSRN electronic journal. 2022. URL: <https://doi.org/10.2139/ssrn.4303230> (date of access: 11.05.2025).
28. Niu Y., Su C., Guo W. Salient object segmentation based on superpixel and background connectivity prior. IEEE access. 2018. T. 6. C. 56170–56183. URL: <https://doi.org/10.1109/access.2018.2873022> (date of access: 11.05.2025).
29. Real-time object segmentation based on convolutional neural network with saliency optimization for picking. Journal of systems engineering

and electronics. 2018. T. 29, № 6. C. 1300. URL: <https://doi.org/10.21629/jsee.2018.06.17> (date of access: 11.05.2025).

30. Robust video object segmentation via propagating seams and matching superpixels / Y. Liang та ін. IEEE access. 2020. T. 8. C. 53766–53776. URL: <https://doi.org/10.1109/access.2020.2981140> (date of access: 11.05.2025).

31. Spatiotemporal activity semantics understanding based on foreground object segmentation: icounter scenario / T.-W. Yu та ін. IEEE access. 2022. C. 1. URL: <https://doi.org/10.1109/access.2022.3178609> (date of access: 11.05.2025).