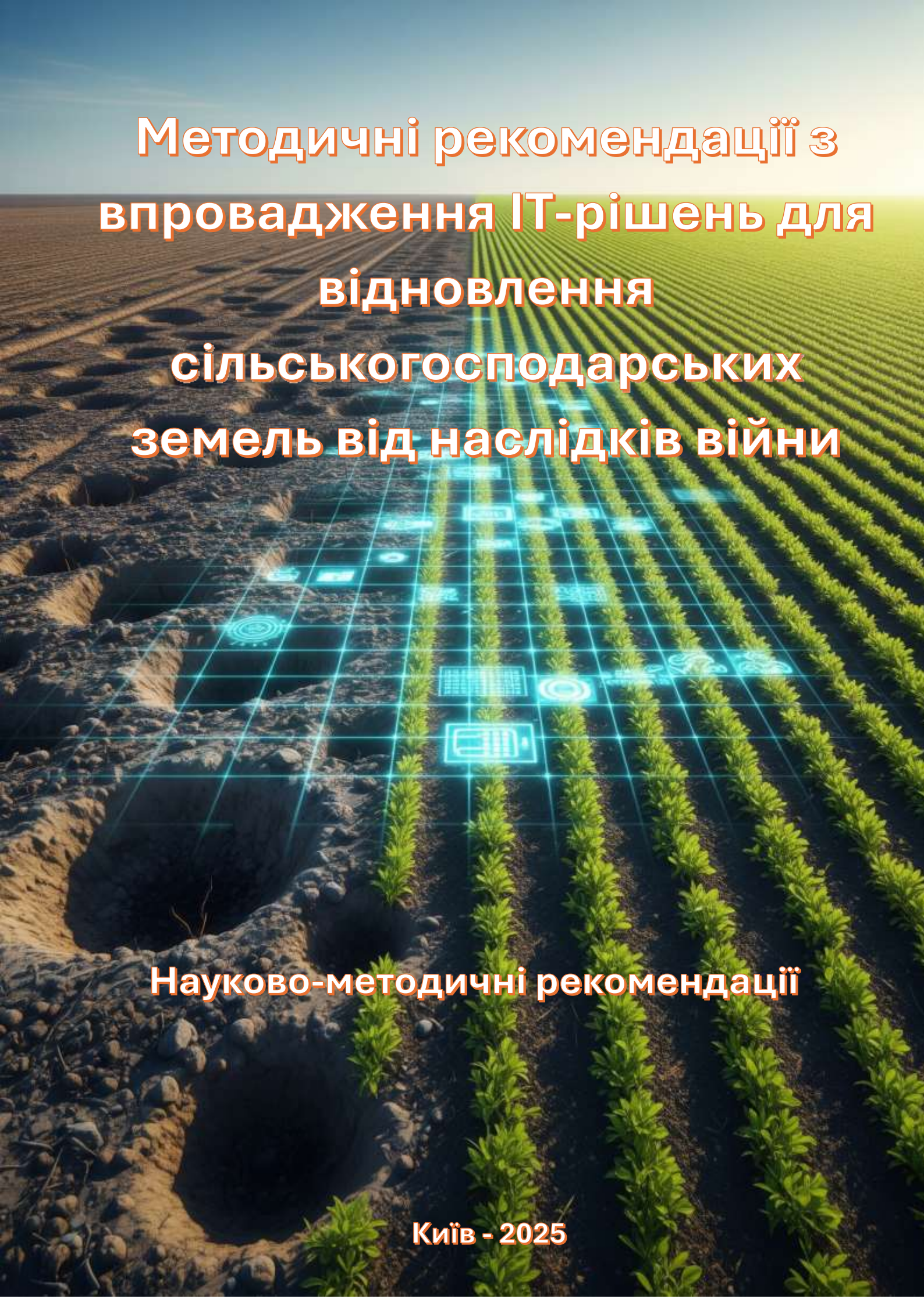


An aerial photograph of a field showing a stark contrast between a damaged, brown, and eroded area on the left and a healthy, green field of crops on the right. A glowing blue digital grid is superimposed over the crops, with various icons floating within it, including a radar, a bar chart, a pie chart, a speech bubble, and a gear, symbolizing the integration of technology in agriculture.

Методичні рекомендації з впровадження ІТ-рішень для відновлення сільськогосподарських земель від наслідків війни

Науково-методичні рекомендації

Київ - 2025

**Національний університет біоресурсів і природокористування України
Міністерства освіти і науки України**

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
З ВПРОВАДЖЕННЯ ІТ-РІШЕНЬ
ДЛЯ ВІДНОВЛЕННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ ЗЕМЕЛЬ
ВІД НАСЛІДКІВ ВІЙНИ**

Науково-методичні рекомендації

Київ - 2025

УДК УДК 631.6:004.9:355.4

Рекомендації розроблені на основі результатів досліджень проведених на кафедрах інформаційних систем і технологій, економічної кібернетики, комп'ютерних наук Національного університету біоресурсів і природокористування України за темою № РН/65-2024 «Центр трансферу технологій штучного інтелекту та відновлення сільськогосподарських земель»

Укладачі: Руденський Р.А., Кравченко В.М., Глазунова О.Г., Тонха О.Л., Саяпін С.П. Болбот І.М.

Рецензенти: Засць Н.А., д.т.н., професор кафедри автоматики та робототехнічних систем ім. акад. І.І. Мартиненка НУБІП України, Клименко Н.А., к.е.н., доцент кафедри економічної кібернетики НУБІП України

Розглянуто і схвалено Вченою радою факультету інформаційних технологій Національного університету біоресурсів і природокористування України від «23» червня 2025 р. протокол № 10

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ З ВПРОВАДЖЕННЯ ІТ РІШЕНЬ ДЛЯ РЕКУЛЬТИВАЦІЇ ЗЕМЕЛЬ ПОРУШЕНИХ ВНАСЛІДОК ВІЙСЬКОВИХ ДІЙ

Підписано до друку _____

Ум. друк. арк. 1,8.

Наклад 100 прим.

Формат 60·80 1/16

Обл. вид. арк. 1

Зам. № _____

ЗМІСТ

ЗМІСТ	4
Вступ.....	5
1. Принципи побудови системи	6
2. Впровадження ІТ рішень для роботи з супутниковими знімками та відео з БПЛА	8
3. Застосування технологій комп'ютерного зору у задачах рекультивації земель	11
4 Застосування технологій QGIS для геоінформаційного аналізу.....	15
5 Методологія планування відновлення земель	28
6 Застосування технологій інтеграції програмних компонентів	30
7 Створення хмарної інфраструктури на AZURE.....	33
Висновки.....	36
Список використаних джерел.....	37

Вступ

Військові дії на території України призвели до масштабних ушкоджень сільськогосподарських угідь, створивши нагальну потребу у розробці ефективних технологічних рішень для швидкого та точного оцінювання стану пошкоджених земель та планування їх відновлення. Традиційні методи наземного обстеження виявилися недостатніми через фактори безпеки, значні масштаби ушкоджень та необхідність оперативного реагування на мінливу ситуацію.

Методичні рекомендації призначені для організацій та установ, що займаються відновленням сільськогосподарських угідь після військових дій, включаючи державні агентства земельних ресурсів, науково-дослідні інститути аграрного профілю, приватні компанії з рекультивації земель та міжнародні організації технічної допомоги. Основними завданнями є надання інструментарію для створення комплексної системи, що забезпечує впровадження автоматизованого підходу до виявлення ушкоджень, планування відновлювальних заходів та моніторингу процесу рекультивації. Актуальність розробки полягає у необхідності переходу від емпіричних методів оцінки ушкоджень до науково обґрунтованих технологічних рішень, що базуються на сучасних досягненнях дистанційного зондування Землі, технологій штучного інтелекту та геоінформаційного аналізу. Інтеграція супутникових даних, технологій комп'ютерного зору та алгоритмів оптимізації дозволяє створити систему, здатну обробляти терабайти геопросторової інформації та надавати об'єктивні рекомендації щодо пріоритетності та послідовності відновлювальних заходів.

Особливістю запропонованого підходу є комплексність рішення, що охоплює весь цикл від автоматизованого виявлення ушкоджень на супутникових знімках до оптимізації логістики рекультиваційних робіт. Методичні рекомендації базуються на принципах модульності, масштабованості, сумісності та безпеки даних, що дозволяє поетапно впроваджувати технологічні рішення відповідно до наявних ресурсів та специфічних потреб організацій.

Практична значущість полягає у можливості скорочення часу виявлення нових ушкоджень, підвищення точності планування відновлювальних робіт та забезпечення об'єктивного моніторингу ефективності заходів рекультивації. Впровадження рекомендованих технологічних рішень сприятиме створенню

єдиної національної платформи моніторингу земельних ресурсів та формуванню бази знань для майбутніх проектів відновлення територій.

1. Принципи побудови системи

Беручи до уваги потенційно великі масштаби системи, складність вирішуваних задач та велику кількість даних, що потребують обробки, базовими принципами, на яких має будуватися система, виділено модульність, масштабованість, інтероперабельність та безпеку даних.

Принцип модульності полягає у створенні системи з взаємопов'язаних, але відносно незалежних функціональних блоків, кожен з яких відповідає за конкретний аспект процесу рекультивації. Сутність даного принципу базується на декомпозиції складної системи на менші підсистеми, що мають чітко визначені інтерфейси взаємодії.

Реалізація модульності досягається через застосування сучасних архітектурних патернів, зокрема мікросервісної архітектури, де кожен модуль функціонує як окремий сервіс з власною базою даних та API. Технічно це забезпечується використанням контейнерних технологій Docker та оркестраторів типу Kubernetes, що дозволяє незалежно розробляти, тестувати та розгортати окремі компоненти системи.

При практичному використанні принцип модульності надає можливість поетапного впровадження функціональності відповідно до наявних ресурсів та пріоритетів, що значно знижує фінансові ризики та дозволяє швидше отримати перші практичні результати. Крім того, модульна архітектура полегшує подальше масштабування системи та інтеграцію нових технологій без необхідності кардинальної перебудови вже існуючих компонентів.

Принцип масштабованості передбачає здатність системи ефективно функціонувати при збільшенні обсягів оброблюваних даних, кількості користувачів або розширенні географічного покриття без деградації продуктивності. Основою цього принципу є архітектурні рішення, що дозволяють як вертикальне масштабування через збільшення потужності серверного обладнання, так і горизонтальне масштабування через додавання нових вузлів до кластера.

Реалізація масштабованості забезпечується використанням хмарних технологій, зокрема платформи Microsoft Azure, що надає можливості автоматичного масштабування ресурсів залежно від поточного навантаження. Технічно це досягається через застосування еластичних сервісів типу Azure Container Instances, Azure Functions та Azure Kubernetes Service, які автоматично адаптують кількість обчислювальних ресурсів до поточних потреб системи. Це в свою чергу дозволяє починати роботу з обмеженими ресурсами на невеликих

територіях та поступово розширювати систему для покриття регіональних або національних масштабів без необхідності кардинальної зміни архітектури.

Принцип сумісності визначає здатність системи до взаємодії з існуючими інформаційними системами, базами даних та програмними продуктами, що вже використовуються або розроблені. Сутність цього принципу полягає у забезпеченні семантичної, синтаксичної та технічної сумісності через використання відкритих стандартів та протоколів обміну даними. Реалізація цього принципу досягається через імплементацію стандартизованих API, зокрема RESTful веб-сервісів, підтримку міжнародних стандартів геопросторових даних OGC (Open Geospatial Consortium) та використання загальноприйнятих форматів даних типу GeoJSON. Технічно це забезпечується створенням адаптерів та трансляторів даних, що дозволяють системі взаємодіяти з різноманітними джерелами інформації, включаючи кадастрові системи, метеорологічні бази даних, системи управління сільськогосподарським виробництвом, що дозволяє максимально використовувати вже наявну IT-інфраструктуру та уникнути дублювання функціональності.

Принцип безпеки даних охоплює комплекс заходів щодо захисту чутливої геопросторової інформації від несанкціонованого доступу, модифікації або знищення, враховуючи стратегічну важливість даних про стан земельних ресурсів та потенційні загрози національній безпеці. Основою цього принципу є багаторівнева система захисту, що включає криптографічне шифрування даних як у стані спокою, так і під час передачі, автентифікацію та авторизацію користувачів, аудит дій та моніторинг безпеки. Реалізація принципу безпеки забезпечується через використання сучасних протоколів шифрування AES-256, SSL/TLS для захищеної передачі даних, багатофакторної автентифікації через Azure Active Directory, системи управління доступом на основі ролей RBAC та технологій zero-trust архітектури. Додатково застосовуються методи диференційної приватності для захисту індивідуальної інформації в агрегованих звітах та технології blockchain для забезпечення цілісності критично важливих даних. У практичному використанні принцип безпеки даних гарантує відповідність вимогам законодавства щодо захисту персональних даних, мінімізує ризики кібератак та витоку інформації, що може бути використана ворожими силами, а також забезпечує довіру громадськості та міжнародних партнерів до результатів роботи системи рекультивації земель.

2. Впровадження ІТ рішень для роботи з супутниковими знімками та відео з БПЛА

Ефективна робота з дистанційними даними становить основу сучасних систем моніторингу стану земельних ресурсів, особливо в умовах післявоєнної рекультивації, де традиційні методи наземного обстеження можуть бути обмеженими через фактори безпеки або масштаби ушкоджень. Інтеграція супутникових даних та відео з безпілотних літальних апаратів створює синергетичний ефект, де супутникові системи забезпечують широкомасштабний моніторинг та виявлення змін на регіональному рівні, тоді як БПЛА надають деталізовану інформацію для локальних ділянок з критично високою роздільністю.

Сучасний ландшафт джерел супутникових даних характеризується значною диверсифікацією, що включає як традиційні наукові місії, так і комерційні платформи, а також новітні API-рішення від технологічних гігантів. Вибір джерел супутникових даних обумовлюється специфікою завдань рекультивації та вимогами до просторової і темпоральної роздільності, при цьому кожна платформа має унікальні переваги та обмеження, що потребують комплексного аналізу для оптимального вибору.

Супутникові системи Sentinel-2 Європейського космічного агентства представляють еталон безкоштовних високоякісних даних дистанційного зондування, забезпечуючи регулярне покриття території з роздільністю 10 метрів у видимому та ближньому інфрачервоному діапазонах з частотою повторного перельоту 5 днів. Основними перевагами Sentinel-2 є відкритий доступ до даних, стандартизована обробка до рівня 2A з атмосферною корекцією, широкий спектральний діапазон з 13 каналами та гарантована довгострокова доступність через європейську космічну політику. Недоліками є обмежена просторова роздільність для детального картування малих об'єктів, відсутність панхроматичного каналу для підвищення роздільності та можливі затримки в доступності даних через завантаженість серверів.

Система Landsat 8/9 НАСА та USGS надає унікальні можливості для аналізу довготривалих трендів через наявність архівних даних, що сягають п'яти десятиліть, що критично важливо для розуміння базового стану території до військових дій та оцінки масштабів змін. Переваги Landsat включають найдовший безперервний архів супутникових даних у світі, стандартизовану методологію обробки та калібрування протягом всього періоду експлуатації, безкоштовний доступ до всіх історичних та сучасних даних, а також високу

радіометричну точність. Обмеженнями є нижча просторова роздільність 30 метрів порівняно з сучасними комерційними системами та більша тривалість циклу повторного перельоту 16 днів.

Google Earth Engine API та супутні сервіси Google Maps Platform представляють революційний підхід до доступу та обробки супутникових даних через хмарні обчислення, надаючи програмний доступ до петабайтів геопросторових даних без необхідності їх локального завантаження та зберігання. Google Earth Engine інтегрує дані Landsat, Sentinel, MODIS та численних інших супутникових місій в єдину платформу з потужними можливостями аналізу часових рядів. Ключовими перевагами є миттєвий доступ до глобальних архівів даних без завантаження, вбудовані алгоритми обробки та аналізу, можливість виконання складних обчислень на серверах Google, інтеграція з машинним навчанням через TensorFlow, а також готові до використання продукти типу Global Forest Change та Global Surface Water.

Google Maps Platform додатково надає доступ до високороздільних зображень через Static Maps API, що включають актуальні супутникові знімки з роздільністю до 60 см для багатьох регіонів світу. Значною перевагою є оперативність оновлення даних, часто випереджаючи традиційні наукові архіви, інтеграція з іншими Google сервісами та можливість отримання зображень у різних форматах та проекціях програмним шляхом.

Комерційні високороздільні супутники типу WorldView, GeoEye та Pleiades забезпечують субметрову роздільність до 30 см, необхідну для точного картування індивідуальних ушкоджень та планування конкретних відновлювальних заходів. Переваги включають найвищу доступну просторову роздільність, можливість замовлення стереопар для створення цифрових моделей рельєфу, гарантовані терміни отримання нових знімків та відсутність хмарності через можливість перепланування зйомки. Основними недоліками є висока вартість, що може сягати сотень доларів за квадратний кілометр, обмежена доступність архівних даних та можливі обмеження доступу для певних регіонів з міркувань безпеки.

Порівняльний аналіз вартості та якості різних джерел даних показує, що для прототипування систем рекультивації земель оптимальним рішенням є використання умовно безкоштовних сервісів Google Maps Platform, що надають доступ до високоякісних зображень з безкоштовними квотами до 28000 завантажень статичних карт на місяць та 25000 завантажень динамічних карт. Це дозволяє розробити та протестувати алгоритми комп'ютерного зору,

налаштувати workflow обробки даних та провести пілотні дослідження без значних фінансових інвестицій.

Таким чином стратегія поетапного переходу від прототипу до промислової системи передбачає початкове використання Google сервісів для відпрацювання алгоритмів та workflow, з подальшим переходом на гібридну модель, що поєднує безкоштовні дані Sentinel-2 та Landsat для регулярного моніторингу з періодичним залученням комерційних високороздільних даних для критичних ділянок. Така стратегія дозволяє збалансувати вартість та якість даних відповідно до специфічних потреб проекту рекультивації та наявного бюджету.

Обробка відеоданих з БПЛА потребує спеціалізованих технологічних рішень, що враховують специфіку польових умов та вимоги до якості результуючої геопросторової інформації. Ключовими технічними вимогами до обладнання є наявність високороздільних камер з матрицею не менше 20 мегапікселів, інтегрованих GNSS-приймачів для забезпечення точної геоприв'язки кожного кадру, та стабілізуючих систем для компенсації вібрацій та нахилів апарату під час польоту. Програмна обробка відеопотоків передбачає застосування фотограмметричних алгоритмів для створення ортофотопланів та цифрових моделей рельєфу, алгоритмів стабілізації для усунення геометричних спотворень, а також систем автоматичного виявлення змін між знімками різних періодів.

Створення єдиного просторово-часового куба даних потребує складних процедур інтеграції різнотипних джерел інформації, що включають геометричну корекцію для приведення всіх знімків до єдиної картографічної проекції, радіометричну нормалізацію для компенсації різниць у спектральних характеристиках різних сенсорів, та темпоральну інтерполяцію для заповнення пропусків у часових рядах спостережень. Ці процедури забезпечують створення консистентного набору даних, придатного для подальшого автоматизованого аналізу методами машинного навчання та комп'ютерного зору.

3. Застосування технологій комп'ютерного зору у задачах рекультивації земель

Застосування технологій комп'ютерного зору у задачах рекультивації земель представляє складну багаторівневу систему, що поєднує класичні методи обробки зображень з сучасними підходами глибокого навчання для автоматизованого виявлення, класифікації та аналізу різнотипних ушкоджень земельного покриву.

Запропонований комплексний підхід дозволяє вирішувати низку завдань для інспекції аграрних угідь на основі супутникових даних, зокрема їх класифікацію, ідентифікацію та локалізацію окремих типів ушкоджень та оцінку рівня ушкоджень на сільськогосподарських землях.

Основною метою створення інтелектуальної системи є автоматизація процедур аналізу великих територій шляхом збору та обробки зображень чи відеозаписів із дронів для оцінки рівня ушкодження сільськогосподарських земель внаслідок військових дій. Особливістю даної предметної області є необхідність роботи з надзвичайно різноманітними типами ушкоджень, що відрізняються за геометричними характеристиками, спектральними властивостями та просторовими патернами, що потребує гібридного підходу до вибору алгоритмів та методів обробки.

Класифікація типів ушкоджень може базуватися на комплексному аналізі геометричних, спектральних та текстурних характеристик об'єктів з використанням сучасних моделей глибокого навчання, втім в межах цих методичних рекомендації перевагу надано дослідженню методів виявлення та детекції кратерів від вибухів, ідентифікації їх розмірів та глибини, що впливає на складність відновлювальних заходів і потребує диференційованого підходу до їх автоматичного виявлення. Малі та “свіжі” кратери зазвичай характеризуються чіткими круглими контурами та високим контрастом з навколишньою територією, що робить їх придатними для виявлення класичними методами контурного аналізу та детекції кіл в OpenCV. Втім більш старі або великі кратери діаметром від 3 до 10 метрів часто мають більш складну внутрішню структуру з можливими вторинними ушкодженнями, що потребує застосування або технологій трансфертного навчання, або тренування власних моделей, наприклад YOLO для локалізації пов'язаних із ними ушкоджень.

Додатковою проблемою дослідження та оцінки рівня деградації сільськогосподарських земель є наявність забудов, доріг, лісів, водних ресурсів,

які покривають значну частину аналізованих територій та ускладнюють роботу моделей комп'ютерного зору.

Для вирішення цього питання розроблена технологічна реалізація системи комп'ютерного зору додатково проводить класифікацію знімків, яка базується на потужних моделях microsoft/Florence-2-large або microsoft/Florence-2-base з автоматичним завантаженням AutoProcessor та AutoModelForCausalLM з Hugging Face Transformers для детального опису зображень та виявлення об'єктів.

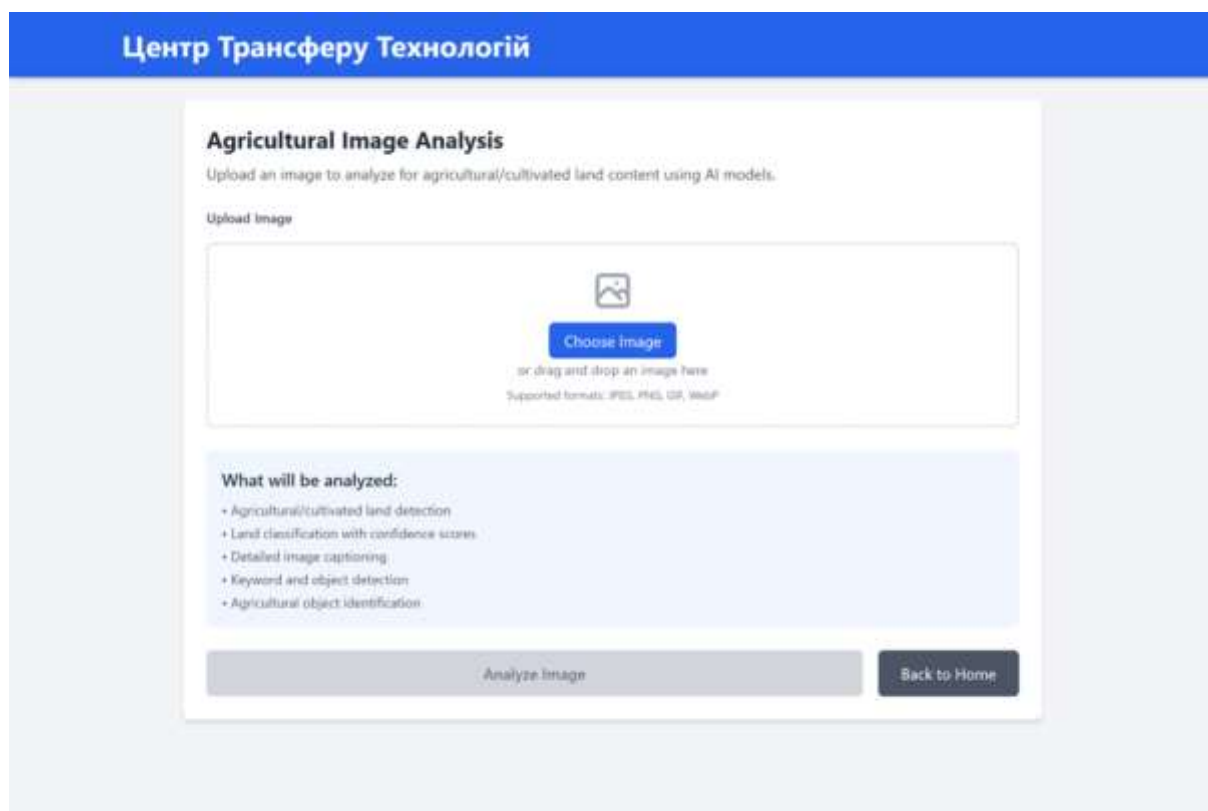


Рисунок 1 – Інтерфейс сервісу аналізу зображень

Модель Florence-2 забезпечує контекстний аналіз сцени з генерацією текстових описів природною мовою, що дозволяє не тільки виявити факт ушкодження, але й проаналізувати його у контексті навколишнього середовища через prompt-и типу <MORE_DETAILED_CAPTION> та <OD> для object detection. У випадку невдачі завантаження Florence-2, передбачено фолбек на модель CLIP (openai/clip-vit-large-patch14) для базової класифікації, що забезпечує стійкість системи до технічних збоїв.

Система визначає землі сільськогосподарського призначення на зображеннях на основі ключових слів у згенерованому описі та виявлених об'єктів, використовуючи семантичний аналіз термінів типу "farm", "field",

"crop", "agricultural land". Це дозволяє автоматично фільтрувати релевантні зображення та фокусувати подальший аналіз на дійсно сільськогосподарських територіях, підвищуючи точність та ефективність системи.

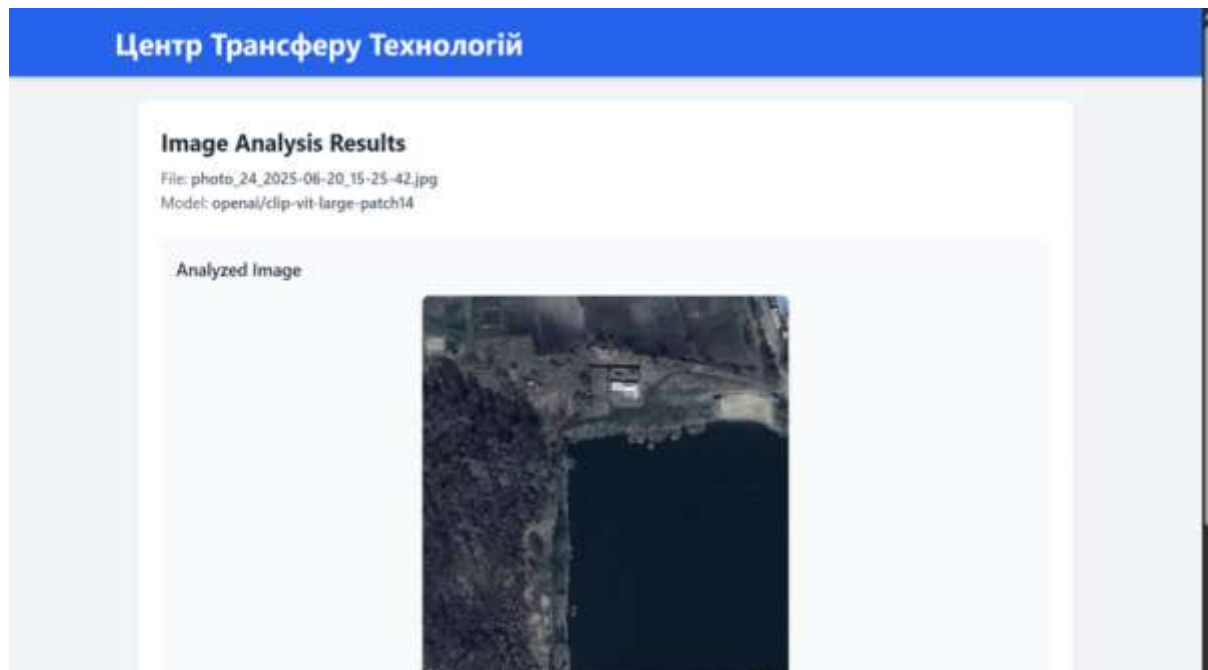


Рисунок 2 – Приклад класифікації зображення

Інтеграція бібліотеки OpenCV для базової обробки зображень забезпечує фундаментальні операції фільтрації шумів, покращення контрасту через адаптивну еквалізацію гістограми, сегментацію за кольором та текстурою через алгоритми кластеризації, а також використання аналізу та оцінки ушкоджень на основі методу детекції контурів та кіл. Ці методи особливо ефективні для попередньої обробки зображень та виявлення простих геометричних форм типу кратерів правильної форми, забезпечуючи швидку первинну оцінку ступеня ушкодження території.

Архітектура YOLO забезпечує швидке виявлення множинних об'єктів на зображеннях великого розміру, що критично важливо при обробці супутникових знімків, що можуть мати розміри в декілька гігапікселів. Передтренована модель YOLO адаптується для локалізації специфічних типів ушкоджень сільськогосподарських земель через навчання на спеціалізованих датасетах з подальшою оптимізацією для роботи на мобільних пристроях та edge-computing пристроях.

Система оптимізації параметрів використовує бібліотеку Optuna для параметричної оптимізації процедур оцінки ушкоджень, що дозволяє автоматично налаштовувати гіперпараметри алгоритмів комп'ютерного зору для

максимізації точності виявлення та мінімізації false positive результатів. Це особливо важливо для адаптації системи до різних географічних регіонів та типів місцевості, де характеристики ушкоджень можуть варіюватися.

4 Застосування технологій QGIS для геоінформаційного аналізу

Геоінформаційна система QGIS відіграє центральну роль у просторовому аналізі даних рекультивації, забезпечуючи інтеграцію результатів автоматизованого виявлення ушкоджень з контекстною геопросторовою інформацією та створення аналітичних продуктів для підтримки прийняття рішень. Особливістю використання QGIS у контексті післявоєнної рекультивації є необхідність роботи з надзвичайно різноманітними типами геоданих, що включають результати дистанційного зондування, дані польових вимірювань, адміністративно-територіальні межі, інформацію про інфраструктурні об'єкти та екологічні обмеження.

Система плагінів QGIS розширює базову функціональність для специфічних потреб аналізу ушкоджень земель. Semi-Automatic Classification Plugin надає можливості для автоматизованої класифікації мультиспектральних супутникових знімків з контрольованим навчанням, що дозволяє виділяти класи земного покриття та виявляти зміни між різними періодами спостережень.

Методики просторового аналізу ушкоджень базуються на комплексному підході, що поєднує геометричний аналіз з атрибутивними характеристиками об'єктів. Створення буферних зон навколо виявлених кратерів дозволяє оцінити вторинний вплив ушкоджень на навколишні території та визначити зони підвищеного ризику для сільськогосподарського використання.

Ключовим нововведенням у аналізі ушкоджень є застосування діаграм Вороного для класифікації ділянок за рівнем хімічного ушкодження та інтеграції цієї інформації з геометричними характеристиками фізичних ушкоджень. Основою цього підходу є визнання того факту, що прямі вимірювання хімічного забруднення ґрунту є надзвичайно дорогими та часозатратними процедурами, особливо на великих територіях, тоді як відновлення кратерів та вирв суттєво залежить як від їх фізичних розмірів, так і від ступеня хімічного забруднення навколишнього середовища.

Методологія використання діаграм Вороного полягає у створенні полігонального розбиття досліджуваної території, де кожна комірка Вороного відповідає зоні впливу однієї точки вимірювання хімічного забруднення, при цьому кожна точка всередині комірки належить до найближчої точки вимірювання (рис. 3).

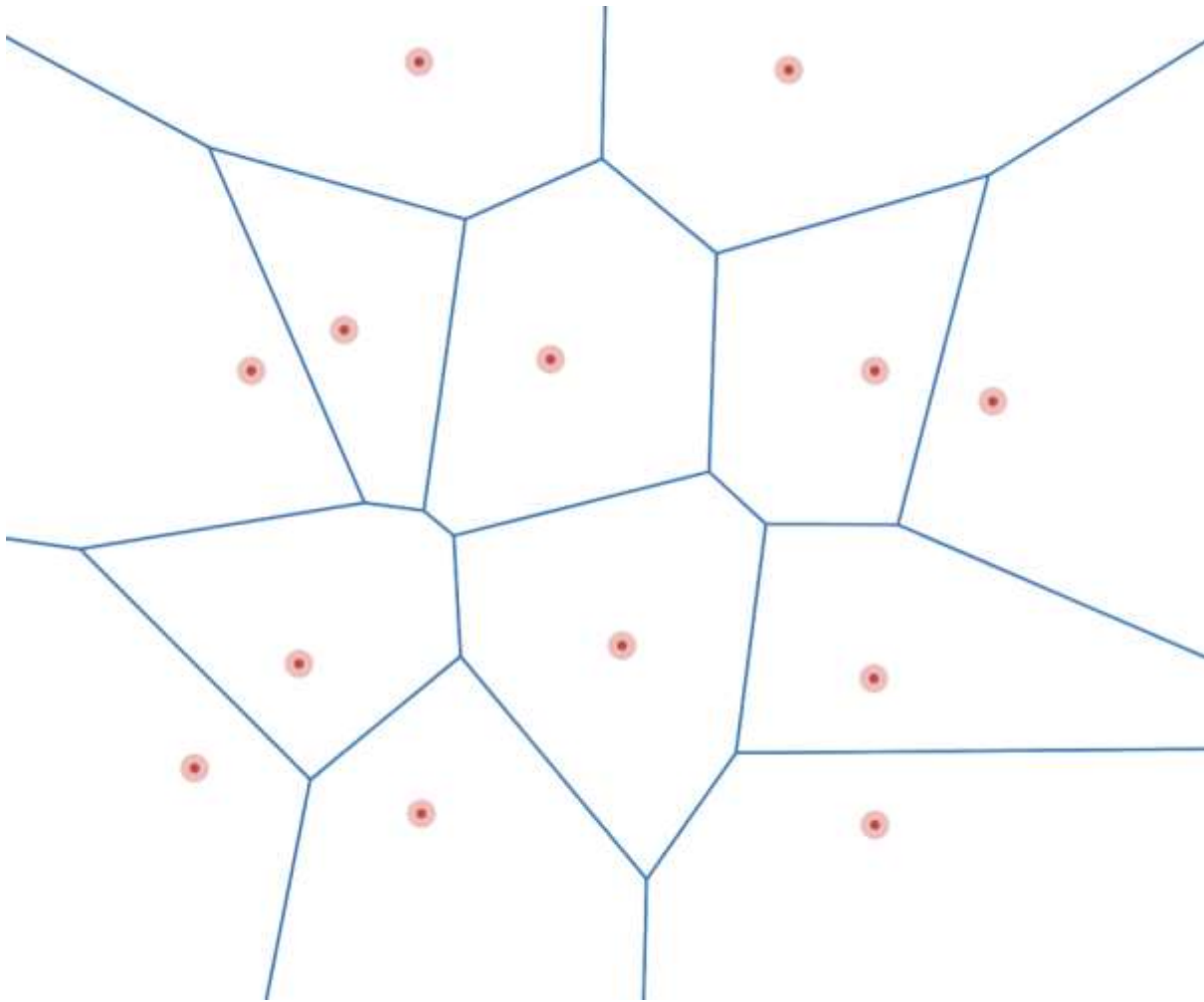


Рисунок 3 – Діаграма Вороного розбиття території за рівнем забруднення

Це дозволяє ефективно інтерполювати рівні забруднення на всю територію на основі обмеженої кількості дорогих хімічних аналізів, проведених у стратегічно обраних локаціях. Кратери та вирви, виявлені методами комп'ютерного зору на супутникових зображеннях, накладаються на діаграму Вороного, що дозволяє кожному ушкодженню присвоїти відповідний рівень хімічного забруднення на основі його просторового розташування.

Інтеграція геометричних характеристик ушкоджень з рівнями хімічного забруднення створює багатовимірну класифікаційну систему складності відновлення. Невеликі кратери у зонах з низьким рівнем забруднення класифікуються як прості для відновлення та потребують стандартних процедур засипання та рекультивації. Великі кратери у зонах високого хімічного забруднення потребують комплексного підходу, що включає попередню детоксикацію ґрунту, спеціальні технології засипання та довгостроковий моніторинг екологічної безпеки. Проміжні категорії ушкоджень отримують відповідні рейтинги складності, що дозволяє оптимізувати розподіл ресурсів та планувати послідовність відновлювальних заходів.

Використання QGIS для планування відновлювальних заходів базується на Network Analyst функціональності для побудови оптимальних маршрутів переміщення спеціалізованої техніки між ділянками з урахуванням стану доріг, топографічних умов та обмежень доступу через безпекові фактори.

Альтернативним підходом до планування рекультиваційних заходів та формування оптимальних маршрутів переміщення спеціалізованої техніки є адаптація задачі комівояжера (Traveling Salesperson Problem, TSP). В цьому випадку розпізнані вирви та їх характеристики визначають набір точок обходу, а час необхідний на рекультивацію пов'язаний із відстанями між вирвами. Таким чином "відстань" в термінах алгоритму представляє собою час, необхідний для виконання комплексу робіт та переміщення техніки і матеріалів до наступної вирви.

Першим кроком планування рекультивації є формування структури даних, що описує виявлені ушкодження та логістичні параметри відновлювальних робіт: відстані, складність, перелік технологічних процедур, матеріалів, техніки, тощо.

```
def create_reclamation_data_model():  
    """Зберігає дані для задачі планування рекультивації."""  
    data = {}  
    # Матриця часових витрат між ушкодженими ділянками  
    # Включає час переміщення техніки + час виконання робіт  
    data["time_matrix"] = [  
        [0, 180, 95, 120, 210, 165, 145, 80, 190, 110, 130, 175, 155],  
        [180, 0, 140, 125, 85, 100, 75, 185, 60, 135, 115, 45, 70],  
        [95, 140, 0, 40, 105, 90, 160, 100, 170, 35, 95, 130, 115],  
        # ... продовження матриці для всіх виявлених ушкоджень  
    ]  
    data["num_teams"] = 1 # Кількість бригад рекультивації  
    data["base_depot"] = 0 # Базова станція (склад матеріалів)  
    return data
```

Матриця часових витрат є двовимірним масивом, де елемент $[i,j]$ представляє загальний час, необхідний для переміщення бригади рекультивації з ділянки i до ділянки j та виконання відновлювальних робіт. Для кожного типу ушкоджень чи сценарію відновлення малі кратери (швидке відновлення), середні кратери (стандартне відновлення), великі кратери (складне

відновлення), зони хімічного забруднення тощо відповідатиме своя матриця відстаней та кількість бригад/техніки, які можуть виконувати відповідні роботи.

Створення моделі маршрутизації

Наступний крок - створення менеджера індексів та моделі маршрутизації, адаптовану для специфіки рекультивації земель:

```
data = create_reclamation_data_model()
manager = pywrapcp.RoutingIndexManager(
    len(data["time_matrix"]), data["num_teams"], data["base_depot"]
)
routing = pywrapcp.RoutingModel(manager)
```

Вхідними параметрами для `RoutingIndexManager` є кількість виявлених ушкоджених ділянок (включаючи базову станцію), кількість бригад рекультивації та номер базової станції, звідки починається та куди повертається маршрут.

Створення функції оцінки часових витрат

Для використання solver-а маршрутизації необхідно створити функцію callback, що приймає будь-яку пару локацій та повертає час, необхідний для переходу між ними з урахуванням специфіки рекультивації:

```
def time_callback(from_index, to_index):
    """Повертає час між двома ділянками."""
    from_node = manager.IndexToNode(from_index)
    to_node = manager.IndexToNode(to_index)
    return data["time_matrix"][from_node][to_node]

transit_callback_index = routing.RegisterTransitCallback(time_callback)
```

Функція callback приймає два індекси та повертає відповідний елемент матриці часових витрат, що включає як час переміщення техніки, так і час виконання рекультивації.

Налаштування параметрів оптимізації

На наступному кроці відбувається пошуку та евристичний метод для знаходження початкового рішення:

```
search_parameters = pywrapcp.DefaultRoutingSearchParameters()
search_parameters.first_solution_strategy = (
```

```
routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC
)
```

Стратегія PATH_CHEAPEST_ARC створює початковий маршрут шляхом послідовного додавання ребер з найменшою вагою, що не ведуть до раніше відвіданих ділянок (окрім базової станції).

Функція виведення результатів

Функція для відображення оптимального плану рекультивації:

```
def print_reclamation_plan(manager, routing, solution):
    """Виводить план рекультивації на консоль."""
    print(f"Загальний час виконання: {solution.ObjectiveValue()} годин")
    index = routing.Start(0)
    plan_output = "Маршрут бригади рекультивації:\n"
    total_time = 0
    while not routing.IsEnd(index):
        plan_output += f" Ділянка_{manager.IndexToNode(index)} ->"
        previous_index = index
        index = solution.Value(routing.NextVar(index))
        total_time += routing.GetArcCostForVehicle(previous_index, index, 0)
    plan_output += f" Базова_станція_{manager.IndexToNode(index)}\n"
    plan_output += f"Загальний час маршруту: {total_time} годин\n"
    print(plan_output)
```

Наступний приклад демонструє застосування адаптованої задачі комівояжера для планування відновлення території площею 10 квадратних кілометрів, на якій методами комп'ютерного зору виявлено 25 кратерів малого розміру та 8 кратерів із зонами хімічного забруднення.

Таким чином дані для задачі складаються з 33 точок на території (25 малих кратерів + 8 кратерів із забрудненням). Програма створює дані у вигляді масиву координат, що відповідають точкам ушкоджень:

```
def create_field_damage_data():
    """Зберігає дані про ушкодження поля."""
    data = {}

    # Координати ушкоджень у блочних одиницях (100м x 100м)
```

```

data["damage_locations"] = [
    (288, 149), (288, 129), (270, 133), (256, 141), (256, 157),
    (246, 157), (236, 169), (228, 169), (228, 161), (220, 169),
    # ... координати всіх виявлених ушкоджень
]
data["num_teams"] = 1
data["depot"] = 0 # Базова станція
return data

```

Обчислення матриці часових витрат

Функція обчислює евклідову відстань між будь-якими двома точками та конвертує її у час, необхідний для рекультивациі, з урахуванням швидкості переміщення техніки та складності робіт (всі часові витрати умовні і мають визначатися за результатами технічної оцінки):

```

def compute_reclamation_time_matrix(locations, damage_types):
    """Створює матрицю часових витрат для рекультивациі."""
    times = {}
    for from_counter, from_node in enumerate(locations):
        times[from_counter] = {}
        for to_counter, to_node in enumerate(locations):
            if from_counter == to_counter:
                times[from_counter][to_counter] = 0
            else:
                # Базовий час переміщення (швидкість 15 км/год)
                travel_time = int(
                    math.hypot((from_node[0] - to_node[0]),
                               (from_node[1] - to_node[1])) * 0.1 / 15 * 60
                )
                # Додавання часу рекультивациі залежно від типу ушкодження
                work_time = get_work_time_by_damage_type(damage_types[to_counter])
                times[from_counter][to_counter] = travel_time + work_time
    return times

def get_work_time_by_damage_type(damage_type):
    """Повертає час рекультивациі залежно від типу ушкодження."""
    work_times = {
        'small_crater': 30, # 30 хвилин

```

```

'medium_crater': 120,      # 2 години
'large_crater': 480,      # 8 годин
'chemical_contamination': 720, # 12 годин
'mixed_damage': 600      # 10 годин
}

return work_times.get(damage_type, 60) # За замовчуванням 1 година

```

При запуску програми вона відображає оптимальний план рекультивації, включаючи послідовність відвідування ушкоджених ділянок та загальний час виконання робіт. OR-Tools знаходить ефективне рішення дуже швидко: менше ніж за секунду на типовому комп'ютері. Загальний час рекультивації території залежить від кількості та складності ушкоджень, наявності матеріалів, кількості бригад та точності часових оцінок. Але у будь-якому разі він дозволяє мінімізувати час використання спеціалізованої техніки та загальний час відновлення сільськогосподарських угідь після військових дій, забезпечуючи науково обґрунтоване планування рекультиваційних заходів.

Процес рекультивації включає очищення від уламків, заповнення вирв ґрунтом або матеріалами, демінування, відновлення родючості ґрунту тощо. *Логістика* в цьому процесі відіграє ключову роль, оскільки передбачає транспортування матеріалів (наприклад, ґрунту, обладнання) до численних пошкоджених ділянок за допомогою автопарку транспортних засобів.

Маршрутизація транспортних засобів (Vehicle Routing Problem – VRP) є узагальненням задачі комівояжера (Traveling Salesman Problem), де кілька транспортних засобів стартують з депо, відвідують локації та повертаються, мінімізуючи критерій, такий як загальна відстань або максимальний час маршруту (Toth, & Vigo, 2014; Laporte, 2009).

Основні варіанти VRP включають:

CVRP (Capacitated VRP) – з обмеженнями на вантажопідйомність транспортних засобів.

VRPTW (VRP with Time Windows) – з часовими вікнами для доставки, що враховує робочі зміни і вікна доступності. Необхідно побудувати маршрут так, щоб прибуття у точку відбувалося у заданий часовий інтервал.

MDVRP (Multi-Depot VRP) – з кількома логістичними центрами (депо) для транспортних засобів і матеріалів. Модель має визначити, з якого депо який транспорт має їхати.

Mixed Fleet VRP – у процесі задіяні різні типи транспортних засобів: вантажівки для матеріалів, спеціалізовані автомобілі для бригад, важка інженерна техніка.

Pickup-and-Delivery – вивіз забрудненого ґрунту на полігон і підвіз чистого ґрунту з кар'єру.

Застосування алгоритмів VRP з використанням відкритого програмного пакета для розв'язання задач оптимізації OR-Tools від Google може суттєво підвищити ефективність координації робіт з відновлення пошкоджених територій:

- скоротити загальний пробіг техніки;
- зменшити час простою обладнання;
- підвищити продуктивність бригад;
- прискорити темпи рекультивації.

Задача маршрутизації транспортних засобів (VRP) є комбінаторною оптимізаційною задачею та задачею цілочисельного програмування, мета якої полягає в знаходженні оптимальних маршрутів для кількох транспортних засобів, що відвідують набір локацій (Official OR Tools).

Процес рекультивації земель після військових дій можна представити як модифіковану VRP, де:

- Локації (вузли):
 - Ділянки з вибуховими вирвами різного розміру
 - Заміновані території різного ступеня забруднення
 - Бази технічних засобів розмінування
 - Пункти утилізації вибухонебезпечних предметів
- Депо: база, де завантажуються земля та матеріали
- Транспортні засоби:
 - Вантажівки або спеціальна техніка з обмеженою місткістю для швидкого та безпечного очищення забрудненої землі
 - Команди саперів з різною спеціалізацією
 - Дрони для попереднього обстеження територій
- Обмеження:
 - Часові вікна для проведення робіт (уникнення темряви або погодних умов)
 - Пріоритетність ділянок (сільськогосподарські землі, критична інфраструктура)

- Технічні можливості транспортних засобів (наприклад, місткість у розмірі 15 м³ ґрунту)
- Безпекові вимоги

При цьому для прийняття рішень щодо рекультивації пошкоджених земельних ділянок треба враховувати, те що кожна вирва має "попит" – об'єм ґрунту, необхідний для заповнення.

Етапи реалізації задачі VRP для рекультивації земельних ділянок:

1. Збір даних:

- Визначення координат пошкоджених ділянок (вирв) за допомогою GIS-систем або супутникових даних.
- Обчислення матриці відстаней (використовуючи Google Distance Matrix API, гіс-системи або Euclidean відстані).
- Визначення попиту (об'єму матеріалів) і місткості транспортних засобів.

2. Моделювання в OR-Tools:

- Застосування Python API OR-Tools.
- Розробка моделі заповнення вирв на підставі вибору одного або декількох варіантів VRP для заповнення вирв.
- Модифікація моделі за рахунок:
 - пріоритезації ділянок і додавання відповідних часових вікон;
 - додавання невизначеності (наприклад, через ризик мін).

3. Розв'язання задачі та аналіз результатів:

- OR Tools генерує набір маршрутів.
- Користувач аналізує отримані маршрути, перевіряє їх на відповідність усім обмеженням і здоровий глузд.
- Результати візуалізуються на мапі (наприклад, за допомогою бібліотек folium чи matplotlib у Python).

4. Інтеграція з іншими інструментами, зокрема: GIS для візуалізації маршрутів.

5. Оцінка ефективності – порівняння з неоптимізованими маршрутами.

Блок створення даних на мові програмування Python має наступний вигляд:

```
def create_data_model():
    data = {}

    data['distance_matrix'] = [...] # Матриця відстаней між вирвами
    data['demands'] = [0, 10, 15, 20, ...] # Попит (об'єм ґрунту для кожної вирви)
```

```

data['vehicle_capacities'] = [15, 15, ...] # Місткість транспортних засобів
data['num_vehicles'] = 3 # Кількість транспортних засобів
data['depot'] = 0 # Депо
return data

```

Наступна функція створює зворотний виклик `distance`, який повертає вартість (відстань, час) переміщення між двома будь-якими локаціями та передає його до розв'язувача. Вона також встановлює вартість дуги, яка визначає вартість переміщення, як відстані дуг:

```

def distance_callback(from_index, to_index):
    """Returns the distance between the two nodes."""

    # Convert from routing variable Index to distance matrix NodeIndex.
    from_node = manager.IndexToNode(from_index)
    to_node = manager.IndexToNode(to_index)

    return data["distance_matrix"][from_node][to_node]
transit_callback_index = routing.RegisterTransitCallback(distance_callback)
routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)

```

Щоб розв'язати VRP, потрібно створити вимір відстані, який обчислює сукупну відстань, пройдену кожним транспортним засобом по його маршруту. Потім можна встановити вартість, пропорційну максимальній загальній відстані по кожному маршруту. Програми маршрутизації використовують виміри для відстеження кількостей, що накопичуються протягом маршруту транспортного засобу. Наведений нижче код створює вимір відстані, використовуючи метод розв'язувача `AddDimension`. Аргументом `transit_callback_index` є індекс для `distance_callback`:

```

dimension_name = "Distance"
routing.AddDimension(
    transit_callback_index,
    0, # no slack
    3000, # vehicle maximum travel distance
    True, # start cumul to zero
    dimension_name,
)
distance_dimension = routing.GetDimensionOrDie(dimension_name)
distance_dimension.SetGlobalSpanCostCoefficient(100)

```

Функція, яка виводить розв'язок:

```
def print_solution(data, manager, routing, solution):  
    """Prints solution on console."""  
  
    print(f"Objective: {solution.ObjectiveValue()}")  
  
    max_route_distance = 0  
  
    for vehicle_id in range(data["num_vehicles"]):  
        if not routing.IsVehicleUsed(solution, vehicle_id):  
            continue  
  
        index = routing.Start(vehicle_id)  
        plan_output = f"Route for vehicle {vehicle_id}:\n"  
        route_distance = 0  
        while not routing.IsEnd(index):  
            plan_output += f" {manager.IndexToNode(index)} -> "  
            previous_index = index  
            index = solution.Value(routing.NextVar(index))  
            route_distance += routing.GetArcCostForVehicle(  
                previous_index, index, vehicle_id  
            )  
  
        plan_output += f"{manager.IndexToNode(index)}\n"  
        plan_output += f"Distance of the route: {route_distance}m\n"  
        print(plan_output)  
        max_route_distance = max(route_distance, max_route_distance)  
    print(f"Maximum of the route distances: {max_route_distance}m")
```

Ця функція відображає маршрути для транспортних засобів та загальну протяжність маршрутів. Фрагмент оптимальної маршрутизації транспортного засобу для заповнення вирв наведено на рис. 4.

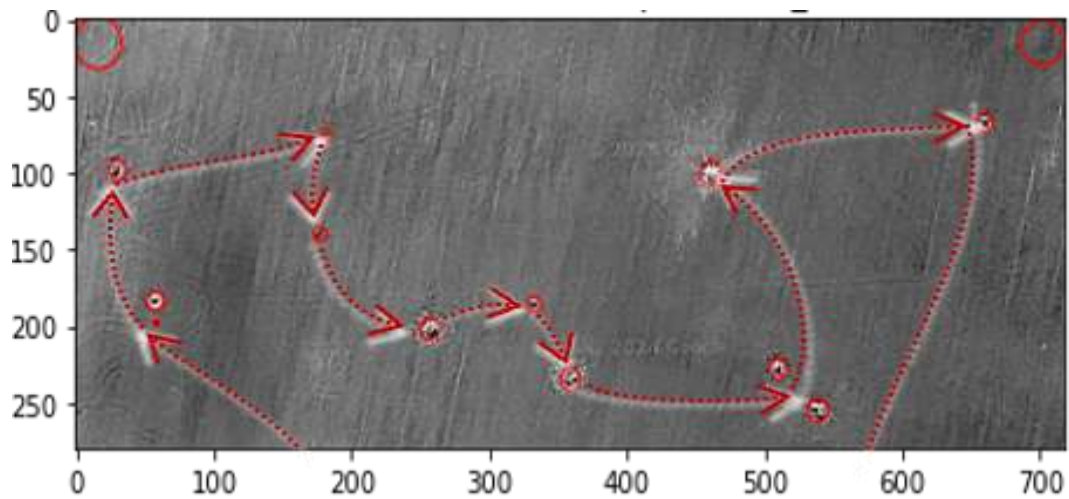


Рисунок 4. – Фрагмент маршруту транспортного засобу
зادля заповнення вирв

Нижче код основної функції класичного варіанту VRP:

```
def main():
    """Entry point of the program."""
    # Створити екземпляр задачі з даними.
    data = create_data_model()
    # Створити менеджер індексу маршрутизації.
    manager = pywrapcp.RoutingIndexManager(
        len(data["distance_matrix"]), data["num_vehicles"], data["depot"]
    )
    # Створити модель маршрутизації.
    routing = pywrapcp.RoutingModel(manager)
    # Створить та зареєструйте зворотний виклик для транзиту.
    transit_callback_index = routing.RegisterTransitCallback(distance_callback)
    # Визначте вартість кожної дуги.
    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)
    # Додати обмеження відстані.
    dimension_name = "Distance"
    routing.AddDimension(
        transit_callback_index,
        0, # no slack
        3000, # максимальна дальність проїзду транспортного засобу
        True, # почати накопичення до нуля
        dimension_name,
```

```

)

distance_dimension = routing.GetDimensionOrDie(dimension_name)
distance_dimension.SetGlobalSpanCostCoefficient(100)
# Встановлення евристики першого рішення.
search_parameters = pywrapcp.DefaultRoutingSearchParameters()
search_parameters.first_solution_strategy = (
    routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC
)

# Розв'язати задачу
solution = routing.SolveWithParameters(search_parameters)

# Вивести рішення задачі на екран.
if solution:
    print_solution(data, manager, routing, solution)
else:
    print("No solution found !")

if __name__ == "__main__":
    main()

```

Зонування території за пріоритетністю відновлення використовує багатокритеріальний аналіз, що враховує екологічну критичність ділянки, її сільськогосподарський потенціал, близькість до населених пунктів та інфраструктурних об'єктів, а також технічну складність відновлювальних заходів.

Аналіз доступності інфраструктури включає розрахунок відстаней до джерел водопостачання, електроенергії, складів будівельних матеріалів та баз технічного обслуговування спеціалізованої техніки. Це дозволяє оцінити логістичні витрати для кожної ділянки відновлення та оптимізувати послідовність робіт для мінімізації транспортних витрат та часу простою обладнання.

5 Методологія планування відновлення земель

Застосування теорії розкладів і, зокрема, адаптація класичної задачі Job Shop Scheduling Problem для планування заходів рекультивації земель представляє інноваційний підхід до оптимізації використання обмежених ресурсів у просторово-розподілених завданнях відновлення. Традиційна постановка Job Shop Problem розглядає планування виконання множини робіт на множині машин з урахуванням технологічних обмежень та послідовності операцій, однак специфіка рекультивації земель потребує суттєвих модифікацій базової моделі для врахування геопросторових факторів, варіативності складності ушкоджень та обмежень реального світу.

Фундаментальною особливістю адаптованої задачі є введення просторової компоненти, де кожна "робота" представляє конкретну ушкоджену ділянку з фіксованими географічними координатами, а "машини" відповідають спеціалізованим бригадам або одиницям техніки з різними можливостями та обмеженнями мобільності. Ця просторова прив'язка створює додаткові обмеження у вигляді часу переміщення між локаціями, що не існує у класичній постановці задачі та потребує розробки спеціалізованих алгоритмів оптимізації, що враховують географічні відстані та стан транспортної інфраструктури.

Класифікація ушкоджених ділянок за складністю відновлення базується на багатофакторній системі оцінювання, що інтегрує результати автоматизованого виявлення ушкоджень методами комп'ютерного зору з даними про хімічне забруднення, отриманими через аналіз діаграм Вороного. Створено чотири основні сценарії складності відновлення. Перший сценарій включає малі кратери до 3 метрів діаметром у зонах з низьким рівнем хімічного забруднення, що потребують стандартних процедур засипання та поверхневої рекультивації з використанням базової техніки типу бульдозерів та культиваторів. Другий сценарій охоплює середні ушкодження або малі кратери у зонах помірного забруднення, що вимагають додаткових заходів детоксикації та використання спеціалізованого обладнання для роботи з забрудненими ґрунтами.

Третій сценарій складності включає великі кратери або ушкодження середнього розміру у зонах високого хімічного забруднення, що потребують комплексного підходу з попередньою детоксикацією, складними земляними роботами та довгостроковим моніторингом.

Четвертий, найскладніший сценарій, охоплює великі кратери у зонах критичного хімічного забруднення, що потребують залучення

високоспеціалізованої техніки, експертних бригад та можливого тимчасового вилучення території з сільськогосподарського обігу на період детоксикації.

Оптимізація використання техніки та людських ресурсів здійснюється через розв'язання модифікованої задачі Job Shop Problem з додатковими обмеженнями та цільовими функціями. Основними типами ресурсів є землерийна техніка різної потужності, транспортні засоби для перевезення ґрунту та будівельних матеріалів, спеціалізоване обладнання для роботи з забрудненими ґрунтами, бригади кваліфікованих робітників різної спеціалізації, а також обмежені запаси матеріалів типу чистого ґрунту, насіння, добрив та детоксикуючих речовин.

Часові обмеження включають сезонні вікна для проведення різних типів робіт, де основні земляні роботи оптимально проводити у сухий період року, сівбу та висадку рослинності у весняно-осінні періоди, а деякі види хімічної обробки ґрунту можуть мати обмеження за температурними умовами. Додатково враховуються погодні умови, що можуть призупинити роботи, та безпекові обмеження, що можуть тимчасово закривати доступ до певних територій.

Цільова функція оптимізації є багатокритеріальною та включає мінімізацію загального часу виконання всіх робіт, мінімізацію логістичних витрат на переміщення техніки та транспортування матеріалів, максимізацію використання наявних ресурсів для уникнення простоїв дорогого обладнання, та максимізацію екологічної ефективності через пріоритизацію відновлення найбільш критичних з екологічної точки зору ділянок. Розв'язання такої складної оптимізаційної задачі потребує застосування метаевристичних алгоритмів, зокрема генетичних алгоритмів для глобального пошуку оптимуму та алгоритмів симульованого відпалення для локальної оптимізації знайдених рішень.ок з координатами та характеристиками

- Множина бригад з різною спеціалізацією та обладнанням
- Часові вікна для виконання робіт (сезонні обмеження)
- Пріоритетність відновлення (критичність ділянки)
- Логістичні обмеження (транспорт, склади матеріалів)

6 Застосування технологій інтеграції програмних компонентів

Створення ефективної системи інтеграції різнорідних компонентів рекультиватії земель базується на сучасній мікросервісній архітектурі, що забезпечує безшовну взаємодію між модулями обробки геопросторових даних, алгоритмами машинного навчання, системами планування та користувацькими інтерфейсами. Комп'ютерна програма розроблена на принципах модульності та підготовлена для переходу на мікросервісну архітектуру, включаючи основний логічний функціонал, що відповідає за різні етапи обробки зображень, бекенд на FastAPI, що забезпечує інтеграцію із окремим React-фронтом, контейнеризованим за допомогою Docker та Nginx.

Центральним елементом такої архітектури є RESTful API, що функціонує як універсальний шлюз для обміну даними та координації робочих процесів між автономними компонентами системи.

Використання принципів мікросервісної архітектури, де кожен функціональний домен представлений окремим сервісом з власною відповідальністю та автономними життєвими циклами розробки і розгортання забезпечує реалізацію наступної логіки:

Сервіс обробки супутникових даних відповідає за прийом, валідацію, попередню обробку та збереження дистанційних даних, надаючи стандартизовані endpoint-и для завантаження нових знімків, запиту метаданих та ініціації процедур атмосферної корекції.

Сервіс комп'ютерного зору інкапсулює алгоритми YOLO, OpenCV та Florence-2, надаючи асинхронні API для виявлення ушкоджень з можливістю відстеження статусу виконання довготривалих обчислювальних задач.

Сервісна архітектура організована навколо спеціалізованого AgriculturalImageAnalysisService, що інкапсулює всю логіку аналізу сільськогосподарських зображень та забезпечує уніфіковані API ендпоінти. Центральний ендпоінт POST /api/analyze-image/ приймає файли зображень, виконує їх класифікацію на предмет сільськогосподарського призначення, надає детальний опис через Florence-2 та виявляє відповідні об'єкти через YOLO, повертаючи структуровані результати у форматі JSON.

Система валідації та стандартизації відповідей реалізована через Pydantic моделі, що забезпечують строгу типізацію та валідацію даних. DetectionResponse модель структурує результати аналізу, включаючи булеві індикатори сільськогосподарського призначення, рівні впевненості, класифікацію, детальні описи, знайдені ключові слова та списки виявлених

об'єктів. `AnalyzeImageResponse` забезпечує загальну обгортку для відповідей API з метаданими про використану модель та статус обробки.

Управління конфігурацією та безпекою здійснюється через централізований `config_loader`, що забезпечує безпечне завантаження API-токенів для Hugging Face та інших зовнішніх сервісів з змінних середовища. Система підтримує гнучке конфігурування через `.env` файли та `.ini` конфігураційні файли, що дозволяє легко адаптувати систему до різних середовищ розгортання.

Паттерн `dependency injection` реалізований через `FastAPI dependencies` модуль, що забезпечує елегантне управління життєвими циклами сервісів та автоматичну обробку ситуацій недоступності сервісів через повернення HTTP 503 статусу. Це дозволяє системі деградувати *gracefully* при проблемах з окремими компонентами.

Фронтенд інтеграція реалізована через повнофункціональний React застосунок з сучасним технологічним стеком, що включає Tailwind CSS для стилізації, Axios для HTTP комунікації та React Router для маршрутизації. Ключовою особливістю є наявність механізму "Mock API" в `apiService.js`, що дозволяє фронтенд розробникам працювати незалежно від стану бекенду, використовуючи заздалегідь визначені мок-дані для всіх типів аналізу.

Асинхронна обробка та `workflow` управління забезпечуються через `event-driven` архітектуру з підтримкою довготривалих операцій машинного навчання. Система відстеження стану включає індикатори завантаження, обробку помилок та автоматичне перенаправлення користувачів на сторінки результатів після завершення аналізу. Компонент `LoadingSpinner` забезпечує візуальний фідбек під час обробки, підвищуючи користувацький досвід.

Контейнеризація та оркестрація системи реалізована через Docker з підтримкою Docker Compose для локального розгортання та готовністю до Kubernetes для продакшн середовищ. Nginx виступає як `reverse proxy` та статичний файловий сервер для React застосунку, забезпечуючи оптимальну продуктивність та масштабованість системи. Архітектура підтримує горизонтальне масштабування через додавання додаткових інстансів сервісів та автоматичний `load balancing`.

Подальший розвиток системи полягає в інтеграції із допоміжними сервісами, які виходять за рамки цього дослідження, але тим не менш формують цілісний методологічний підхід. Геоаналітичний сервіс - підтримує функціональність QGIS через PyQGIS API та забезпечує просторовий аналіз,

включаючи побудову діаграм Вороного, розрахунок буферних зон та багатокритеріальний аналіз пріоритетності відновлення.

Сервіс планування реалізує алгоритми розв'язання адаптованої Job Shop Problem і надає API для оптимізації розкладів робіт з урахуванням ресурсних та часових обмежень.

Кожен сервіс дотримується принципів REST архітектури, використовуючи стандартні HTTP методи GET, POST, PUT, DELETE для відповідних операцій читання, створення, оновлення та видалення ресурсів.

Стандартизація форматів обміну даними здійснюється через використання JSON для структурованих даних та GeoJSON для геопросторових об'єктів, що забезпечує сумісність з широким спектром геоінформаційних систем та веб-клієнтів. Метадані супутникових знімків структуруються, що полегшує інтеграцію з існуючими каталогами геоданих та забезпечує довгострокову збереженість інформації. Результати автоматизованого виявлення ушкоджень кодуються у форматі, сумісному з Open Geospatial Consortium стандартами, що дозволяє імпорт в різноманітні ГІС платформи без додаткових перетворень.

Система обробки подій та workflow базується на асинхронній архітектурі з використанням черг повідомлень для забезпечення надійності та масштабованості обробки великих обсягів геоданих. Надходження нового супутникового знімка автоматично ініціює каскад обробних процедур, включаючи валідацію якості даних, геометричну корекцію, виявлення змін порівняно з попередніми знімками, автоматизоване виявлення ушкоджень та оновлення просторової бази даних. Кожен етап workflow генерує події, що логуються для аудиту та можуть використовуватися для налаштування оповіщень та тригерів подальших процесів.

Система моніторингу та відстеження стану інтегрує метрики продуктивності окремих сервісів з бізнес-метриками процесу рекультивациі, забезпечуючи комплексне розуміння ефективності системи як з технічної, так і з предметної точки зору. Автоматизовані алерти налаштовуються для критичних сценаріїв, включаючи виявлення нових масштабних ушкоджень, аномальні затримки в обробці даних, та досягнення порогових значень завантаженості обчислювальних ресурсів.

7 Створення хмарної інфраструктури на AZURE

Розгортання хмарної інфраструктури на платформі Microsoft Azure для підтримки процесів рекультивації земель вимагає комплексного підходу до архітектурного планування, що враховує специфічні вимоги до обробки великих обсягів геопросторових даних, забезпечення високої доступності сервісів та гарантування безпеки чутливої інформації про стан територій. Фундаментальною основою такої інфраструктури є побудова багаторівневої архітектури, що поєднує сервіси зберігання, обчислень, машинного навчання та аналітики у єдину інтегровану екосистему.

Resources

Recent Favorite	
Name	Type
 crater_detection_AML	Azure Machine Learning workspace
 craterdetectionstorage	Storage account
 crater_detection	Resource group
 crater-detection-api	App Service
 craterDetectionFunctionApp	Function App
 crater-detection-web-app	Static Web App
 NorthEuropeLinuxDynamicPlan	App Service plan
 Machine Learning	Subscription
 craterDetectionFunctionApp	Application Insights
 craterDetectionCR	Container registry

Рисунок 5. – Ресурси для побудови системи

Архітектура зберігання даних базується на ієрархічному підході, де Azure Blob Storage функціонує як центральне сховище для raw-даних супутникових знімків, відеофайлів з БПЛА та архівних матеріалів, забезпечуючи практично необмежену масштабованість та різні рівні доступу для оптимізації витрат на зберігання рідковживаних архівних даних. Azure Data Lake Storage Gen2 надає можливості для структурованого зберігання великих обсягів геоданих з підтримкою ієрархічної файлової системи та оптимізацією для аналітичних навантажень.

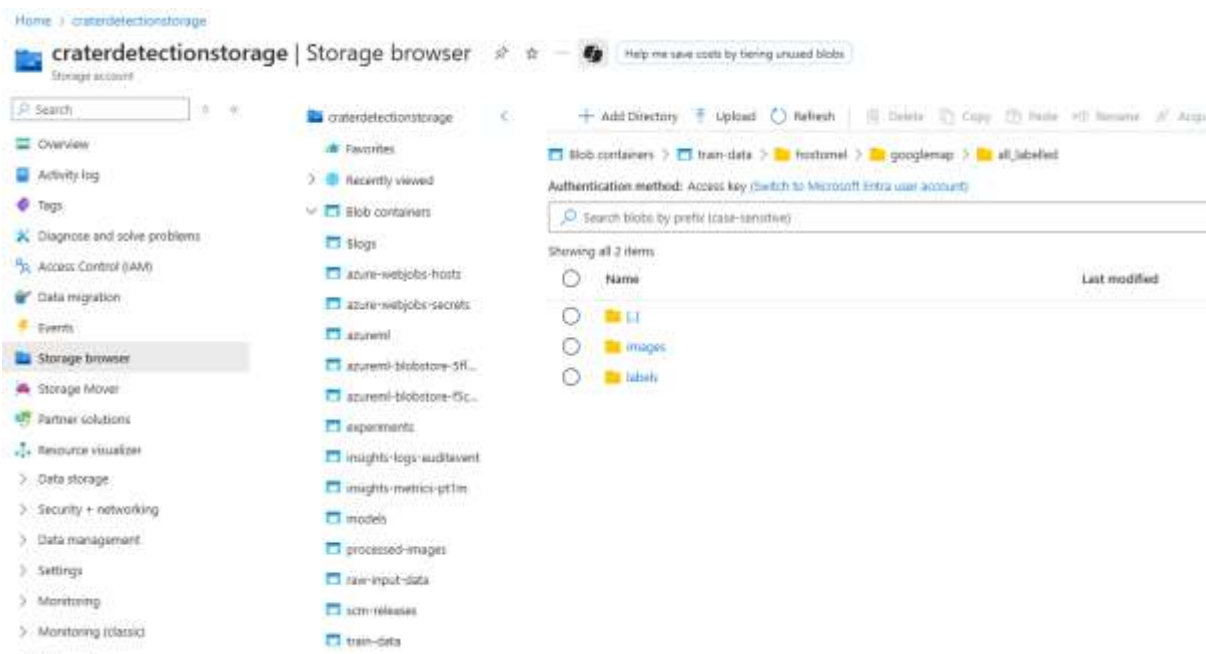


Рисунок 6. – Фрагмент структури сховища даних

Реляційні метадані та результати аналізу можуть зберігатися в Azure SQL Database з географічною реплікацією для забезпечення високої доступності та можливості відновлення після аварій, втім на етапі побудови прототипу така функціональність може бути зайвою.

Обчислювальна інфраструктура спроектована як гібридна система, що поєднує постійно працюючі сервіси для базової функціональності з еластичними обчислювальними ресурсами для пікових навантажень обробки. Azure Kubernetes Service забезпечує оркестрацію мікросервісної архітектури з автоматичним масштабуванням під час надходження великих пакетів супутникових даних. Azure Functions реалізують serverless обробку тригерів та легких обчислювальних задач, що дозволяє мінімізувати витрати на обчислювальні ресурси під час періодів низької активності. Для ресурсоємних задач машинного навчання використовуються Azure Machine Learning compute clusters з підтримкою GPU прискорення для навчання моделей комп'ютерного зору.

Інфраструктура машинного навчання інтегрує Azure Machine Learning Studio як центральну платформу для розробки, навчання та розгортання ML-моделей з підтримкою повного життєвого циклу від експериментального дослідження до продакшн deployment. Azure Cognitive Services, зокрема Computer Vision API, надають готові до використання можливості розпізнавання образів, що можуть використовуватися як baseline для порівняння з спеціалізованими моделями. Azure Custom Vision дозволяє швидко

створювати та налаштовувати моделі класифікації для специфічних типів ушкоджень без глибокого знання машинного навчання.

Система безпеки та контролю доступу базується на багаторівневій архітектурі захисту з використанням Azure Active Directory як центральної системи ідентифікації та аутентифікації. Role-Based Access Control забезпечує гранульований контроль доступу до різних компонентів системи відповідно до функціональних обов'язків користувачів. Azure Key Vault централізовано управляє криптографічними ключами, сертифікатами та секретами, забезпечуючи їх безпечне зберігання та ротацію. Мережева безпека реалізується через Azure Virtual Networks з сегментацією трафіку, Network Security Groups для контролю доступу на рівні мережі та Azure Firewall для захисту від зовнішніх загроз.

Архітектура моніторингу та логування забезпечує комплексне відстеження стану всіх компонентів системи через Azure Monitor, що агрегує метрики продуктивності, журнали подій та пользовательські метрики у єдину аналітичну платформу. Application Insights надає детальну аналітику продуктивності веб-додатків та API з автоматичним виявленням аномалій та bottleneck-ів. Log Analytics централізовано збирає та аналізує журнали з усіх компонентів системи, дозволяючи створювати складні запити для виявлення паттернів та потенційних проблем.

Система резервного копіювання та відновлення після аварій включає автоматизоване резервне копіювання критичних даних в Azure Backup з різними політиками retention для різних типів інформації. Географічна реплікація забезпечує збереження копій даних у різних Azure регіонах для захисту від регіональних аварій. Azure Site Recovery надає можливості швидкого відновлення функціональності системи у резервному дата-центрі з мінімальним RTO та RPO.

Оптимізація витрат досягається через використання Azure Cost Management для постійного моніторингу та аналізу витрат на хмарні ресурси, автоматизованих правил зупинки непотрібних ресурсів у неробочий час, використання Reserved Instances для постійно працюючих сервісів та Spot Instances для пакетної обробки даних, що не є критичною до часу виконання.

Висновки

Представлені методичні рекомендації формують комплексну технологічну основу для ефективного впровадження інформаційних технологій у процеси рекультивації земель, пошкоджених внаслідок військових дій. Розроблений підхід базується на інтеграції сучасних технологій дистанційного зондування, штучного інтелекту та геоінформаційного аналізу, що дозволяє автоматизувати ключові етапи оцінювання стану пошкоджених територій та планування відновлювальних заходів.

Ключові наукові та технологічні досягнення полягають у наступному :

Обґрунтовано застосування принципів модульності, масштабованості, сумісності та безпеки даних як фундаментальних основ побудови системи рекультивації земель. Модульна архітектура забезпечує можливість поетапного впровадження функціональності відповідно до наявних ресурсів, тоді як принцип масштабованості дозволяє ефективно адаптувати систему від локальних проектів до національних програм відновлення. Сумісність гарантує можливість інтеграції з існуючими інформаційними системами, а багаторівнева система безпеки захищає стратегічно важливу геопросторову інформацію.

Розроблено комплексну методологію роботи з різнотипними джерелами дистанційних даних, що включає супутникові системи, API-сервіси Google Earth Engine. Запропонована стратегія поетапного переходу від прототипування на безкоштовних сервісах до промислової експлуатації гібридних рішень дозволяє оптимізувати співвідношення вартості та якості даних відповідно до специфічних потреб проектів рекультивації.

Створено інноваційний підхід до автоматизованого виявлення ушкоджень земель через інтеграцію класичних методів OpenCV з сучасними моделями глибокого навчання YOLO та Florence-2. Система забезпечує не лише виявлення та локалізацію кратерів різного розміру, але й контекстний аналіз земель сільськогосподарського призначення, що підвищує точність та ефективність подальшого аналізу.

Запропоновано використання діаграм Вороного для інтеграції результатів автоматизованого виявлення фізичних ушкоджень та інформації щодо класифікації територій за рівнем хімічного забруднення. Такий підхід дозволяє ефективно інтерполювати дорогі хімічні аналізи на всю територію та створювати багатовимірну систему класифікації складності відновлення.

Практичні результати та переваги впровадження полягають у адаптації методів дослідження операцій, зокрема задачі комівояжера та проблеми

маршрутизації транспортних засобів для специфіки планування рекультиваційних заходів, що дозволяє оптимізувати використання спеціалізованої техніки та мінімізувати логістичні витрати. Розроблені алгоритми враховують просторові обмеження, варіативність складності ушкоджень та сезонні фактори виконання робіт.

Створено архітектуру RESTful API на базі FastAPI з підтримкою мікросервісної організації та контейнеризації через Docker, що забезпечує гнучкість розгортання та масштабування системи. Інтеграція з React-фронтом та механізмом Mock API дозволяє паралельну розробку компонентів та швидке прототипування функціональності.

Розроблено комплексну стратегію розгортання хмарної інфраструктури на платформі Microsoft Azure з оптимізацією витрат через використання еластичних сервісів та автоматизованого масштабування. Багаторівнева система безпеки та резервного копіювання гарантує надійність зберігання критично важливої геопросторової інформації.

В результаті розробки та впровадження системи очікується скорочення часу виявлення нових ушкоджень з тижнів до днів через автоматизовані системи моніторингу супутникових даних та оперативну обробку зображень з БПЛА. Підвищення точності планування відновлювальних робіт на 30-40% завдяки науково обґрунтованому підходу до класифікації ушкоджень та оптимізації розподілу ресурсів. Забезпечення об'єктивного моніторингу ефективності рекультиваційних заходів через регулярний аналіз супутникових даних та порівняння з базовими показниками стану території до військових дій. Створення єдиної бази знань для майбутніх проектів відновлення територій та формування наукового підґрунтя для державної політики у сфері рекультивації земель.

Перспективи подальшого розвитку вбачаються в інтеграції з IoT-сенсорами для безперервного моніторингу стану ґрунту та екологічних параметрів відновлюваних територій. Використання передових технологій штучного інтелекту для прогнозування успішності різних сценаріїв рекультивації та адаптації до змін клімату.

Розширення функціональності системи для роботи з різними типами природних та техногенних катастроф, що дозволить використовувати розроблені рішення у більш широкому контексті управління земельними ресурсами. Створення міжнародної платформи обміну досвідом та технологіями у сфері рекультивації земель після військових конфліктів.

Методичні рекомендації становлять значний внесок у розвиток сучасних підходів до відновлення пошкоджених територій та можуть служити основою для створення національної системи моніторингу та управління земельними ресурсами в умовах постконфліктного відновлення. Впровадження запропонованих технологічних рішень сприятиме прискоренню процесів відновлення сільськогосподарського виробництва та забезпеченню продовольчої безпеки держави.

Список використаних джерел

1. Bey, A., Sánchez-Paus Díaz, A., Maniatis, D., et al. (2016). Collect Earth: Land use and land cover assessment through augmented visual interpretation. *Remote Sensing*, 8(10), 807.
2. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.
3. Burns, B., & Beda, J. (2019). *Kubernetes: up and running: dive into the future of infrastructure*. O'Reilly Media.
4. Chen, L., Zhang, H., Xiao, J., et al. (2017). SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5659-5667.
5. Copernicus Programme. (2023). Sentinel-2 Mission Guide. European Space Agency. <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>
6. Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3), 399-409.
7. Google Developers. (2023). Traveling Salesperson Problem (TSP). OR-Tools Documentation. <https://developers.google.com/optimization/routing/tsp>
8. Google OR-Tools Team. (2023). OR-Tools User's Manual. <https://developers.google.com/optimization>
9. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.
10. Howse, J. (2013). *OpenCV Computer Vision with Python*. Packt Publishing.
11. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics. <https://github.com/ultralytics/ultralytics>
12. Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5), 778-782.
13. Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing.
14. Li, Y., Zhang, H., Xue, X., et al. (2018). Deep learning for remote sensing image classification: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6), e1264.
15. Ma, L., Liu, Y., Zhang, X., et al. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166-177.
16. Masse, M. (2011). *REST API design rulebook: designing consistent RESTful web service interfaces*. O'Reilly Media.

17. Microsoft Corporation. (2023). Azure Machine Learning Documentation.
<https://docs.microsoft.com/en-us/azure/machine-learning/>
 18. OpenCV Team. (2023). OpenCV Documentation: Image Processing.
https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html
 19. Pekel, J. F., Cottam, A., Gorelick, N., & Belward, A. S. (2016). High-resolution mapping of global surface water and its long-term changes. *Nature*, 540(7633), 418-422.
 20. Perron, L., & Furnon, V. (2019). OR-Tools. <https://github.com/google/or-tools>
 21. QGIS Development Team. (2023). QGIS User Guide.
https://docs.qgis.org/3.28/en/docs/user_manual/
 22. Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. *Proceedings of the International Conference on Machine Learning*, 8748-8763.
 23. Ramirez, S. (2023). FastAPI Documentation: Tutorial - User Guide.
<https://fastapi.tiangolo.com/tutorial/>
 24. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.
 25. Richardson, L., & Ruby, S. (2007). *RESTful web services*. O'Reilly Media.
 26. Sherman, G. (2014). *The PyQGIS programmer's guide: extending QGIS with Python*. Locate Press.
 27. Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
 28. Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: Problems, methods, and applications* (2nd ed.). Society for Industrial and Applied Mathematics.
 29. Van Hentenryck, P., & Michel, L. (2005). *Constraint-based local search*. MIT Press.
 30. Zhang, L., Zhang, L., & Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 22-40.
- Zhu, Z., Wulder, M. A., Roy, D. P., et al. (2019). Benefits of the free and open Landsat data policy. *Remote Sensing of Environment*, 224, 382-385.

Підписано до друку _____
Ум. друк арк. 2,5
Наклад 100 прим.

Формат 60x84\16

Зам № _____

Віддруковано у редакційно видавничому відділі НУБіП України
вул. Героїв Оборони 15, Київ, 03041
тел. 527-84-55