

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

**НАУКОВО-МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ІЗ ЗАСТОСУВАННЯ ШІ
ДЛЯ ОПЕРАТИВНОГО МОНІТОРИНГУ
ПОШКОДЖЕНИХ ЗЕМЕЛЬ**

Київ – 2024

УДК [519.6+004.8+004.9]:631

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

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

Рецензенти: С.І. Осадчий, д.т.н., професор кафедри конструкції повітряних суден, авіадвигунів та підтримання льотної придатності Української державної льотної академії, В.М. Смолій, д.т.н., професор кафедри інформаційних систем і технологій Національний університет біоресурсів і природокористування України.

Розглянуто і схвалено Вченою радою факультету інформаційних технологій Національного університету біоресурсів і природокористування України від «19» грудня 2024 р. протокол № 5

НАУКОВО-МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ІЗ ЗАСТОСУВАННЯ ШІ ДЛЯ ОПЕРАТИВНОГО МОНІТОРИНГУ ПОШКОДЖЕНИХ ЗЕМЕЛЬ

Підписано до друку _____
Ум. друк. арк. 1,8.
Наклад 100 прим.

Формат 60·80 1/16
Обл. вид. арк. 1
Зам. № _____

ЗМІСТ

ВСТУП	4
1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ВИКОРИСТАННЯ ІШІ ДЛЯ ОЦІНКИ ПОШКОДЖЕНЬ ЗЕМЕЛЬНИХ РЕСУРСІВ	5
2. МОЖЛИВОСТІ АДАПТАЦІЇ(ТРАНСФЕРТНЕ НАВЧАННЯ) НА ПРИКЛАДІ МОДЕЛІ РУСДА	7
3. МЕТОДИКА НАВЧАННЯ МОДЕЛЕЙ КОМП'ЮТЕРНОГО ЗОРУ НА БАЗІ ЗІБРАНИХ ТА РОЗМІЧЕНИХ ДАНИХ	9
4. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЕВРИСТИЧНИХ АЛГОРИТМІВ ДЛЯ ДЕТЕКЦІЇ КОНТУРІВ	13
5. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕДУРИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ	17
6. ОЦІНКА РЕСУРСІВ ДЛЯ РЕКУЛЬТИВАЦІЇ ДЕГРАДОВАНИХ ТЕРИТОРІЙ	22
ВИСНОВКИ	23
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	24
ДОДАТКИ	25

ВСТУП

Тривалі військові дії на сході та півдні України призвели до значних руйнувань природного середовища, які торкнулися як інфраструктури, так і земельних ресурсів. Масштабні втрати, спричинені бойовими діями, ставлять перед державою комплекс завдань, що включають як ліквідацію наслідків руйнувань, так і відновлення сільськогосподарських територій. Земельний фонд зазнав значних пошкоджень, включаючи утворення вирв від вибухів, зміну рельєфу, забруднення важкими металами та токсичними речовинами.

Оцінка масштабу цих пошкоджень є першим і критично важливим етапом у розробці ефективної стратегії рекультивації земель. Цей процес вимагає використання сучасних технологій, які поєднують дистанційне зондування, аерофотозйомку та методи штучного інтелекту (ШІ). У цьому контексті штучний інтелект стає потужним інструментом для аналізу супутникових знімків і даних, отриманих із дронів, забезпечуючи швидке і точне виявлення та класифікацію пошкоджень.

Головна *мета* цих рекомендацій — сприяти формуванню комплексного підходу до оцінки деградації земельних ресурсів, використовуючи передові методи ШІ. Це забезпечить надійність і об'єктивність оцінки, зокрема ідентифікацію вирв, обчислення їх параметрів та інтеграцію цих даних у плани відновлення земель.

Для досягнення цієї мети запропоновано багатоетапну процедуру, для кожного етапу якої визначені послідовні кроки та інструменти штучного інтелекту для їх реалізації рис. 1.

1. Збір та попередня обробка даних	2. Розмітка та підготовка даних	3. Вибір архітектури та навчання моделей	4. Оцінка обсягів деградованих земель	5. Картографування деградації земель	6. Оцінка ресурсів для рекультивації
Інструментарій: Google API Кроки: - Визначення координат та розмірів ROI - Визначення підрегіонів на основі заданого рівня масштабування - Збір супутникових зображень, що покривають весь ROI	Інструментарій: Label Studio, VGG Image Annotator - Визначення координат, розмірів та типів кратерів - Аналіз розподілів розмірів вирв - Аугментація даних - Фільтрація та генерування додаткових ознак	Досліджувані підходи та архітектури: - YOLO (You Only Look Once) - Faster R-CNN - SSD (Single Shot Detector) - OpenCV - Unsupervised підходи - Класифікація	Оцінка об'ємів кратерів на основі виявлених розмірів (глибина і діаметр). Оцінка обсягу ґрунту, необхідного для відновлювальних заходів	Інструменти: QGIS, ArcGIS (накладання виявлених кратерів на супутникові зображення) Результат: карта деградації	Аналіз історичних даних про типи ґрунтів, потреби рослинності та кліматичні умови регіону для інформування про вимоги до ресурсів Розробка моделі розподілу ресурсів, яка враховує доступні матеріали, робочу силу та часові обмеження.

Рис. 1 Етапи процедури детекції та оцінки розмірів вирв від снарядів за супутниковими зображеннями

1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ВИКОРИСТАННЯ ШІ ДЛЯ ОЦІНКИ ПОШКОДЖЕНЬ ЗЕМЕЛЬНИХ РЕСУРСІВ

Запропонований підхід базується на інтеграції сучасних технологій дистанційного зондування, аерофотозйомки та алгоритмів штучного інтелекту для досягнення високої точності оцінки пошкоджень земель. Основними аргументами на користь цього підходу є такі:

- *Ефективність супутникових даних.* Супутникові знімки дозволяють отримати повну картину великих зон ураження, виявляючи значні зміни рельєфу, такі як кратери та вирви. Висока частота знімків і детальність спектрального аналізу дозволяють відстежувати зміни у часі.

- *Деталізація даних за допомогою дронів.* Аерофотозйомка з використанням дронів забезпечує деталізацію в локальних зонах. Це дозволяє ідентифікувати навіть дрібні пошкодження, які залишаються непомітними на супутникових знімках. Крім того, дрони дозволяють створювати тривимірні моделі рельєфу, враховуючи глибину та об'єм вирв.

- *Застосування алгоритмів комп'ютерного зору та машинного навчання.* Технології ШІ автоматизують процес обробки даних, зокрема класифікацію пошкоджень і визначення їх масштабів. Згорткові нейронні мережі дозволяють аналізувати текстурні, тіньові та спектральні характеристики зображень, що забезпечує високу точність виявлення вирв.

- *Адаптивність і масштабованість.* Запропонований підхід може бути адаптований до різних умов і масштабів ураження. Інтеграція різних джерел даних (супутники, дрони) дозволяє враховувати специфіку кожного регіону.

- *Економічна доцільність.* Використання автоматизованих методів ШІ значно знижує витрати на ручну обробку даних, забезпечуючи оперативність і високу якість оцінки пошкоджень.

Ці технології дозволяють сформувати об'єктивну базу даних для оцінки шкоди і подальшого планування рекультиваційних заходів. Запропонований підхід відповідає сучасним вимогам моніторингу та забезпечує комплексний аналіз стану земель, уражених внаслідок військових дій.

Першим кроком запропонованого підходу, як і будь-якого проекту із машинного навчання є збір супутникових зображень регіону інтересу (ROI) за допомогою API Google. Цей етап включає визначення координат та розмірів ROI із подальшим визначенням центрів підрегіонів для збору супутникових зображень заданого масштабу та адекватною роздільною здатністю для виявлення кратерів.

Для збору даних на цьому етапі дослідження використовували Google API, а сама процедура збору передбачала реалізацію наступних кроків:

1. Визначити координати регіона для збору даних. При цьому у якості регіону було обрано **Гостомельський район Київської області**. Ділянка 100 кв. км. із координатами(дивіться додаток А):

lat_min=50.555378719704066,
lon_min=30.180589491227035,
lat_max=50.64531088029594,
lon_max=30.322276108772964

2. Визначити рівень наближення (Zoom level) та розбити регіон на підрегіони відповідно до обраного рівня наближення. В межах цього дослідження використовувались наступні рівні:

Zoom 17: На цьому рівні можна побачити дороги, будівлі, але не всі деталі місцевості. Роздільна здатність зображень на цьому рівні приблизно 0.3 метра на піксель ($1 / 3,000$), що для зображення 640x640 пікселів дає приблизні лінійні розміри: $640 \times 0.3 \approx 192$ метри (точні значення дещо відрізняються для різних значень широт).

Zoom 18: Це більш детальний рівень, на якому видно більше об'єктів, таких як номери будинків та деталі інфраструктури. Роздільна здатність зображень на цьому рівні приблизно 0.15 метра на піксель ($1 / 6,000$). Для зображення 640x640 пікселів це дає: $640 \times 0.15 \approx 96$ метрів.

Zoom 19: Це один з найвищих рівнів масштабування, який надає ще більше деталей. Роздільна здатність зображень на цьому рівні приблизно 0.075 метра на піксель ($1 / 12,000$). Для зображення 640x640 пікселів лінійні розміри складатимуть: $640 \times 0.075 \approx 48$ метрів.

Відповідно до визначених рівнів та координат та розміру регіону було зібрано наступні дані для кожного рівня:

Zoom 17 – 441 зображень

Zoom 18 – 1764 зображень

Zoom 19 – 6889 зображень

Для розмітки даних використовували Label Studio для задачі object detection. В результаті отримали набори розмічених даних у форматі Yolo, в якому ідентифіковано воронки від снарядів та вибухів на представлених наборах фото знімків(дивіться додаток Б).

Подальше дослідження включало проведення наступних експериментів:

- *Трансфертне навчання*, тобто адаптація моделей навчених на детекцію кратерів планетарних об'єктів для розпізнавання вирв.

- Навчання моделей комп'ютерного зору на базі зібраних та розмічених даних
- Використанням *евристичних алгоритмів* детекції контурів із наступною ідентифікацією кіл та пошук оптимальних гіперпараметрів для алгоритмів.

Перевагою останнього підходу є те що він не потребує навчання складних моделей, а передбачає лише параметризацію евристичних фільтрів. Таким чином, на відміну від глибокого навчання моделей комп'ютерного зору ми використовуємо алгоритми оптимізації гіперпараметрів.

2. МОЖЛИВОСТІ АДАПТАЦІЇ(ТРАНСФЕРТНЕ НАВЧАННЯ) НА ПРИКЛАДІ МОДЕЛІ PYCDA

Модель, реалізована в модулі PyCDA орієнтована на автоматичне визначення кратерів на Марсі та Місяці. В основі алгоритму лежать конволюційні нейронні мережі. Теоретичні аспекти алгоритму представлені в роботі [1]. В роботах [2,3] досліджені питання та технології автоматичного розпізнавання та підрахунку кількості кратерів, але жоден з цих методів не став повністю автоматизованим чи стандартним інструментом у аналізі планетарних поверхонь.

Підходи до алгоритмів виявлення кратерів (CDA) включають методи аналізу зображень, такі як виявлення країв і трансформація Хафа. Однак у останні роки деякі з них також включили машинне навчання, включаючи архітектури нейронних мереж.

Вибір модуля PyCDA обумовлений тим, що він надає автоматизоване рішення на основі машинного навчання для проблеми виявлення та підрахунку кратерів, яке використовує згорткову нейронну мережу (CNN) з даними реального зразка (тренувальними даними), наданими існуючою базою даних марсіанських кратерів з діаметрами понад 1 км [4,5]. Цей алгоритм заснований на використанні згорткової нейронної мережі (CNN) та TensorBox, відкритої платформи для детекції об'єктів на базі Google TensorFlow. Для тренування та аналізу CDA були використані мозаїки термальних емісійних зображень USGS Astrogeology (THEMIS), що покривають всю екваторіальну широтну смугу Марсу. Кожне мозаїчне зображення охоплює область приблизно 2700 км на 1800 км. Мозаїки були розділені на 6387 тайлів, кожен з яких має ширину 1280 пікселів і висоту 960 пікселів. Подальше вдосконалення алгоритму для зображень з високою роздільною здатністю, дозволило виявляти кратери від 100м, що в термінах виявлення кратерів від вирв означає, що алгоритм CDA здатний виявити широкий спектр кратерів, які різняться за розміром і виглядом. CDA виявила 80% кратерів, ідентифікованих у режимі ручної ідентифікації. При цьому CDA виявила

деякі кратери, які були упущені ручною ідентифікацією, і навпаки. Таким чином, точність CDA досягає рівня точності, порівнянного з ручним виявленням.

Таким чином, PyCDA – це готовий модуль який реалізує алгоритм для детекції кратерів (CDA), написаний на Python. Він націлений на те, щоб зробити дослідження CDA модульним і зручним у використанні.

Для адаптації цього алгоритму до задачі виявлення вирв було досліджено та апробовано різні методи підготовки зображень, спрямовані на те, щоб зробити їх найбільш схожими на зображення на яких навчалась модель CDA.

Для цього використовувалась наступна процедура:

1. Переведення зображення в відтінки сірого
2. Зменшення шуму на зображеннях за допомогою Гаусівського фільтру
3. Адаптація яркості і контрастності зображення за допомогою 2d конволюційної фільтрації (рис. 2)

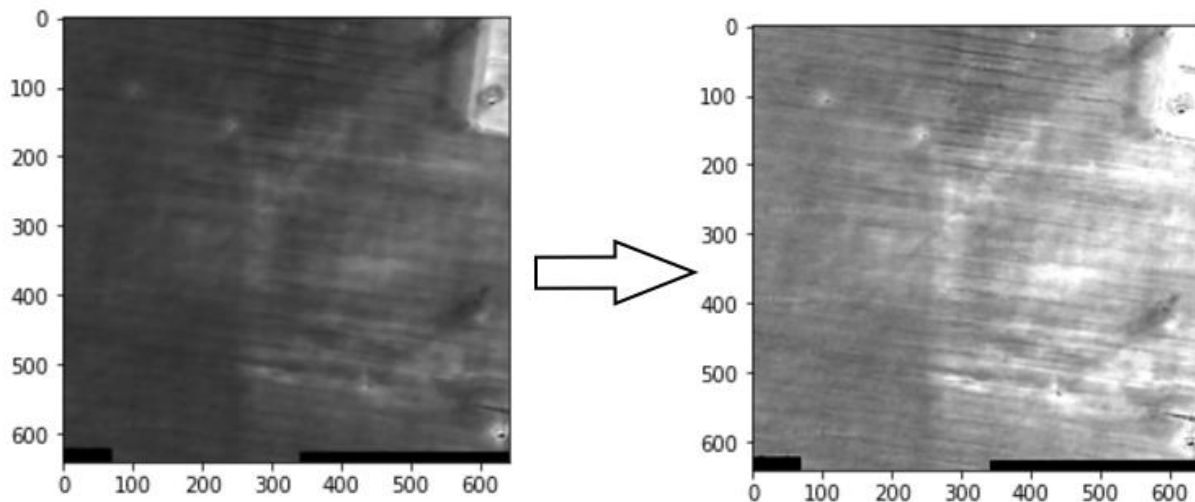


Рис. 2 Адаптація яркості і контрастності зображення за допомогою 2d конволюційної фільтрації

Приклад реалізації детекції із застосуванням цього підходу показано на рис.3

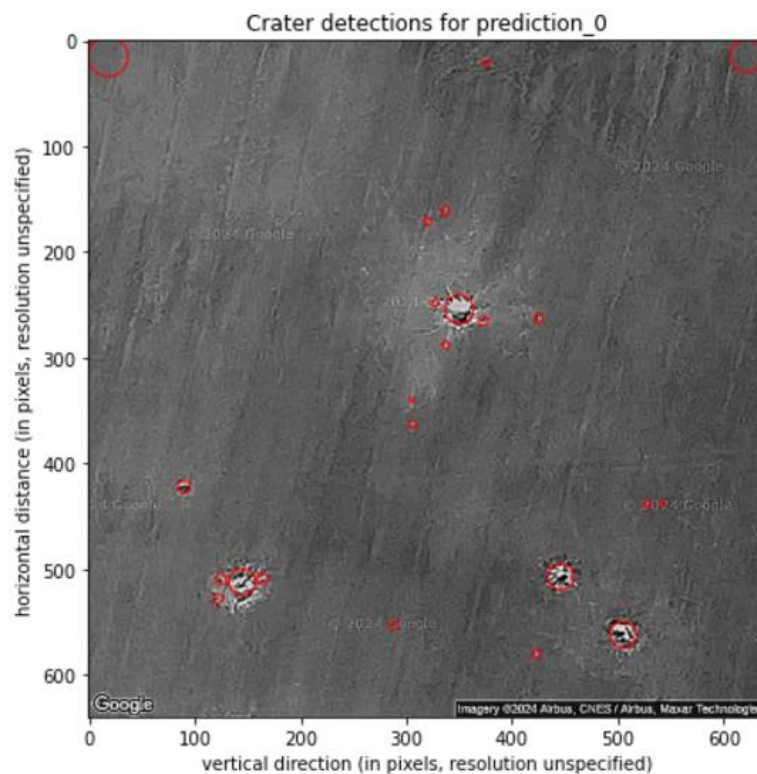


Рис.3 Детекція із використанням алгоритму PyCDA

3. МЕТОДИКА НАВЧАННЯ МОДЕЛЕЙ КОМП'ЮТЕРНОГО ЗОРУ НА БАЗІ ЗІБРАНИХ ТА РОЗМІЧЕНИХ ДАНИХ

Використання згорткових нейронних мереж для вирішення подібних задач зазвичай дозволяє досягти більшої точності у виявленні кратерів [2]. Проведені експерименти включали дослідження таких архітектур як

YOLO (You Only Look Once), Faster R-CNN (із різними базовими моделями) та SSD (Single Shot Detector).

Розглянемо базову архітектуру YOLOv5, як основну модель, що використовувалась для визначення вирв на супутникових зображеннях. Загалом архітектура складається із трьох компонент (рис. 4):

1. Основа або «Хребет» (Backbone) — ця частина моделі складається із конволюційних шарів та відповідає за формування набору ознак для детекції об'єктів на зображенні

2. Мережа об'єднання ознак (Neck) – ця частина моделі трансформує отримані ознаки в три карти ознак (P3-P5, рис. 4)

3. Предиктивні шари (Head) – в цій частині моделі вже відбувається визначення границь об'єктів та їх класів.

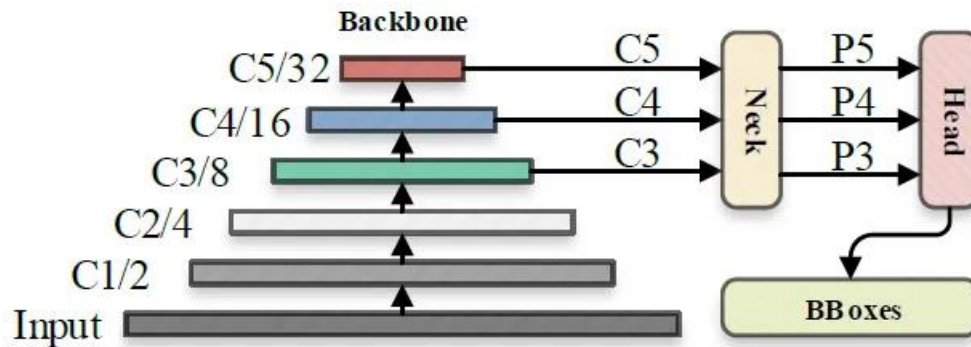


Рис. 4 Загальна архітектура моделі YOLOv5

Загальна процедура обробки зображення відбувається наступним чином: зображення проходить через вхідний шар (input), де відбувається його масштабування та підготовка і надсилається до шарів Основи, де відбувається формування ознак різного розміру.

Отримані карти ознак різного розміру за допомогою мережі об'єднання ознак (neck) об'єднуються у три карти ознак P3, P4 та P5 (в YOLOv5 розміри виражені у величинах 80×80 , 40×40 та 20×20) для виявлення малих, середніх і великих об'єктів на зображенні відповідно. Після того, три карти ознак надсилаються на предиктивні шари (head), де обчислюються оцінки впевненості моделі та межі знайдених кратерів для кожного пікселя в карті ознак за допомогою попередньо визначеного анкера, щоб отримати багатовимірний масив, що включає клас об'єкта, довіру до класу, координати рамки, ширину та висоту.

Встановивши відповідні порогові значення (confthreshold, objthreshold) для фільтрації непотрібної інформації в масиві та виконавши процес ненадмірного подавлення (NMS), можна отримати фінальну інформацію про виявлення. Більш детальна структура шарів моделі YOLOv5 представлена на рис. 5

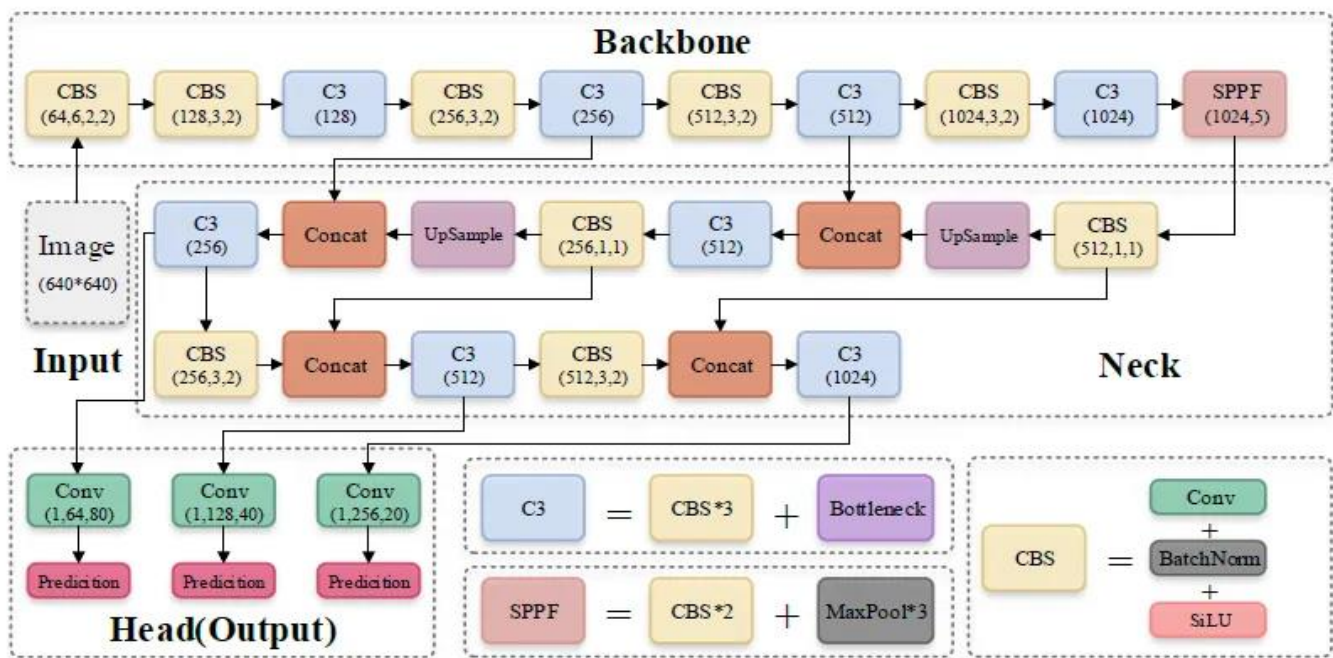
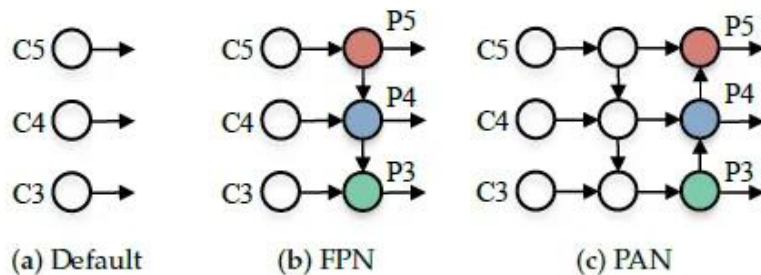


Рис. 5. Структура шарів моделі YOLOv5

Як можна побачити із рис 5, базовим компонентом архітектури є додавання кількох модулів CBS (Convolution + BatchNorm + SiLU) та модулів C3, і на завершення підключається один модуль SPPF.

Модулі CBS у поєднанні із модулями C3 сприяють формуванню ознак, тоді як модуль SPPF підвищує здатність основної структури до вираження ознак.

Додатково YOLOv5 використовує методи FPN та PAN для підготовки карти ознак. Основна ідея FPN полягає в збільшенні виходу карти ознак (C3, C4 та C5), створеної за допомогою кількох операцій зменшення розміру за допомогою згорток з



мережі видобутку ознак, для отримання кількох нових карт ознак (P3, P4 та P5), призначених для виявлення об'єктів різних розмірів (рис. 6).

Рис. 6 Модифікації мережі об'єднання ознак (Neck)

Останні, предиктивні шари (Head) моделі вже відповідають за визначення границь об'єктів та їх класів.

В цій частині відбувається визначення центрів та розмірів об'єктів. При цьому значення координат верхнього лівого кута зображення встановлюється на (0, 0), а правого нижнього — (1, 1). Координати центрів, висоти та ширини об'єктів визначаються у відносних значеннях. Наприклад центр зображення відповідатиме координатам (0.5, 0.5)

Для навчання моделі та підбору гіперпараметрів використовувались наступні стратегії:

Підготовка та аугментація даних — в межах цієї стратегії ми використовували дані різної роздільної здатності (zoom 17, zoom 18, змішані дані); різні підходи до аугментації даних (повороти, відображення, зміна контрасту, яскравості, насиченості кольорів, перемішування сегментів зображень, масштабування, тощо).

Підготовка та валідація міток — в межах цієї стратегії ми валідували розміри та положення міток, видаляли замалі мітки, які не мали практичної цінності та не впливали суттєво на загальну площу уражену вирвами.

Зміна параметрів голови моделі, зокрема перехід до анкерів із більш збалансованим відношенням сторін, що відповідає природі вирв, які зазвичай мають майже квадратну форму.

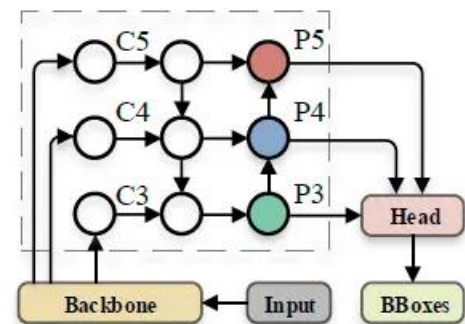
Зміна розміру моделі — в межах цієї стратегії при проведенні експериментів щодо детекції вирв використовувались наступні версії архітектури моделі YOLOv5:

YOLOv5n - ця модель має 40 шарів та 1.9 мільйона параметрів, для дотренування використовувалась модель, яка вже була навчена на даних по місячним кратерам, але донавчання моделі на зібраних даних приводило до перенавчання, що свідчило про недостатню складність моделі для цієї задачі.

YOLOv5s – ця модель має 50 шарів та близько 7.2 мільйона параметрів. Вона показала більшу точність та достатню швидкість навчання.

YOLOv5m – яка має 66 шарів та близько 21.6 мільйона параметрів. Ця версія дозволила отримати найкращі результати з точки зору точності із достатньою швидкістю.

YOLOv5l – незважаючи на те, що ця модель більш складна точність отриманих результатів дещо впала, при цьому зниження точності спостерігалось як в процесі тренування так і валідації, що свідчить про надмірну складність моделі для зібраних даних, її гіршу спроможність навчатися на них та необхідність розширення наборів даних та їх варіабельності для отримання кращих результатів.



Виходячи із отриманих результатів використання найбільш складної моделі YOLOv5x вбачається недоцільним для зібраних даних.

Базові налаштування та приклад застосування моделі YOLOv5m представлено на рис. 7.

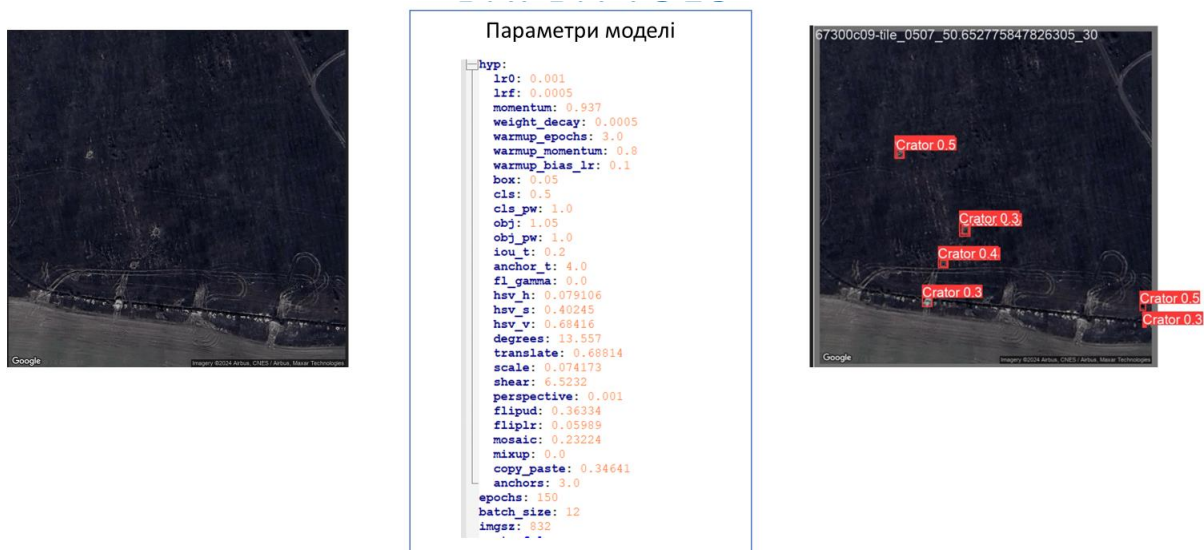


Рис. 7 Налаштування та приклад застосування моделі YOLOv5m

4. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЕВРИСТИЧНИХ АЛГОРИТМІВ ДЛЯ ДЕТЕКЦІЇ КОНТУРІВ

Наступним підходом, який було апробовано для розпізнавання вирв став підхід, заснований на використанні евристичних алгоритмів детекції контурів із наступною ідентифікацією кіл та пошук оптимальних гіперпараметрів для алгоритмів.

Детектор контурів Кенні це метод виявлення контурів, який використовує багатоступінчастий алгоритм для виявлення широкого спектру контурів на зображеннях. Він був розроблений Джоном Ф. Кенні в 1986 році. Кенні також створив обчислювальну теорію виявлення контурів, яка пояснює, чому ця техніка працює.

Цей метод не лише дозволяє отримувати корисну структурну інформацію з зображень, але й значно зменшує обсяг даних, які потрібно обробити. Він широко застосовується в різних системах комп'ютерного зору. Кенні виявив, що вимоги до застосування виявлення контурів в різноманітних системах зору є відносно схожими. Таким чином, рішення для виявлення контурів, що відповідає цим вимогам, може бути реалізовано в широкому діапазоні ситуацій. Цей метод забезпечує виявлення контурів

з низьким рівнем помилок, та гарантує що означає кожен заданий контур на зображенні має позначатися лише один раз, а там, де це можливо, шум на зображенні не повинен створювати помилкових контурів.

Процес виявлення контурів за алгоритмом Кенні можна розділити на п'ять різних етапів:

- Застосування Гаусівського фільтра для згладжування зображення з метою видалення шуму.
- Визначення градієнтів інтенсивності зображення.
- Застосування порогового значення градієнта або подавлення нижньої межі для усунення спірних реакцій на виявлення контурів.
- Застосування подвійного порогу для визначення потенційних контурів.
- Відстеження контурів за допомогою гістерезису: фіналізація виявлення контурів шляхом подавлення всіх інших слабких контурів, які не з'єднані з сильними контурами.

Оскільки результати виявлення контурів легко підлягають впливу шуму на зображенні, важливо фільтрувати шум, щоб запобігти помилковому виявленню, викликаному цим. Для згладжування зображення до нього застосовується Гаусівське фільтрувальне ядро.

Вибір розміру Гаусівського ядра впливає на ефективність згладжування. Чим більший розмір, тим вища спроможність фільтра прибирати шуми. З помітним збільшенням розміру Гаусівського фільтра збільшується і локалізаційна помилка при виявленні контурів. Типовим значенням, яке використовується для більшості випадків є розмір фільтра 5×5 , проте це також може змінюватись залежно від конкретних ситуацій.

Контур на зображенні може бути орієнтований в різні напрямки, тому алгоритм Кенні використовує чотири фільтри для виявлення горизонтальних, вертикальних та діагональних контурів на розмитому зображенні. Оператор виявлення контурів (в реалізації `opencv` – оператор Собеля) повертає значення для першої похідної в горизонтальному напрямку (G_x) та у вертикальному напрямку (G_y). На базі цих даних можна визначається градієнт та його напрямок

Наступний крок алгоритму Кенні – це подавлення нижнього порогу, або мінімальне подавлення величин градієнтів, що є технікою звуження контурів. Цей підхід дозволяє виявити місця з найрізкішими змінами значення інтенсивності на зображенні.

Після застосування не максимального подавлення залишкові пікселі контурів надають більш точне представлення реальних контурів на зображенні. Однак деякі з пікселів залишаються, викликані шумом і змінюваністю кольору. Щоб врахувати ці спірні реакції, важливо фільтрувати пікселі з низькою градієнтною величиною та

зберігати пікселі контурів з високою градієнтною величиною. Це досягається шляхом вибору високих і низьких порогових значень. Якщо градієнтна величина пікселя краю перевищує високий поріг, він позначається як сильний піксель краю. Якщо градієнтна величина пікселя краю менша за високий поріг, але більша за низький поріг, він позначається як слабкий піксель краю. Якщо ж градієнтна величина пікселя краю менша за низький поріг, вона подавляється. Обидва порогові значення визначаються експериментально і залежатимуть від змісту конкретного вхідного зображення.

Сильні пікселі контурів обов'язково повинні бути включені у фінальне зображення; вони вважаються такими, що походять з істинних контурів на зображенні. Проте існує деяка невизначеність щодо слабких пікселів контурів. Необхідно визначити, чи ці пікселі походять з істинного контуру чи є результатом шуму або змін кольору.

Цей алгоритм використовує ідею, що слабкі пікселі контурів, пов'язані з істинними краями, зазвичай будуть з'єднані з сильним пікселем контурів, тоді як реакції шуму залишаться незв'язаними. Для відстеження з'єднання контурів застосовують аналіз об'єктів, перевіряючи слабкий піксель контуру та його сусіднє середовище. Якщо в блоці є хоча б один сильний піксель контуру, цей слабкий піксель можна вважати таким, що підлягає збереженню. Ці слабкі пікселі контурів стають сильними, що, у свою чергу, може призвести до збереження сусідніх слабких пікселів.

В бібліотеці OpenCV для ідентифікації контурів використовується функція `cv2.Canny()`. Основними аргументами цієї функції є значення фільтрів для відстеження контурів за допомогою гістерезису, а саме `minVal` і `maxVal` відповідно. Також додатково можна змінювати розмір ядра Собеля, яке використовується для знаходження градієнтів зображення. Ще одним додатковим параметром є `L2 gradient`, який визначає порядок норми для знаходження величини градієнта. Підтримуються або перший або другий порядок. За результатами проведених експериментів перші два параметри показали найбільшу чутливість алгоритма до виділення контурів, тому для зменшення розміру простору пошуку оптимальних параметрів для двох останніх використовувались типові значення.

Приклад ідентифікації контурів для зібраних супутникових зображень представлений на рис. 8. Він відображає результати зменшення шуму та детекції контурів.

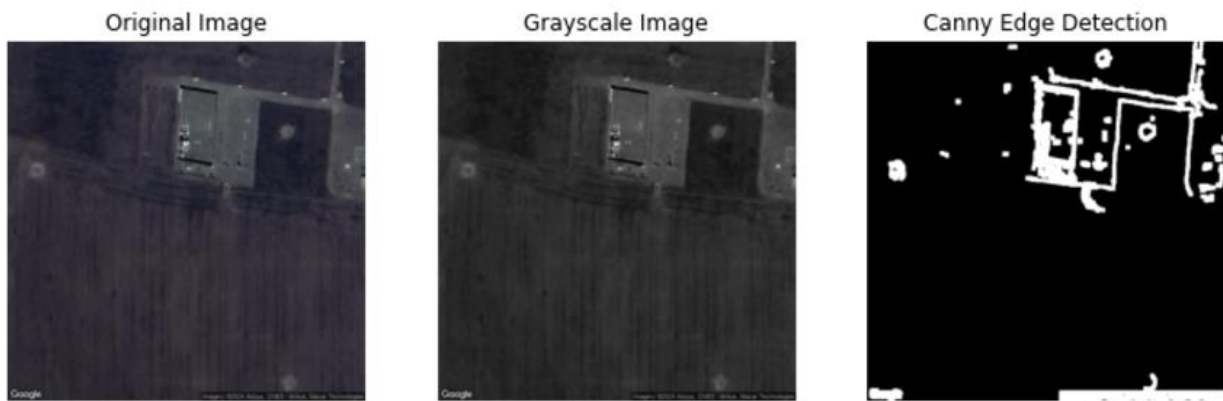


Рис. 8 Результати детекції контурів.

Наступним кроком запропонованого підходу є аналіз виявлених контурів та ідентифікація кіл. Для цього використовували перетворення Хафа для кіл. Це перетворення зображення, яке дозволяє виділяти круглі об'єкти із зображення, навіть якщо вони частково приховані або неповні. Це як правило, ігнорує видовжені еліпси та ефективно шукає об'єкти з високим ступенем радіальної симетрії, причому кожен ступінь симетрії отримує один «голос» у пошуковому просторі. Під час пошуку перетворення може вимірювати центр і радіус кожного кругового об'єкта на зображенні.

Головна ідея полягає в тому, що кожне коло може бути описане трьома параметрами: координатами центру (x , y) та радіусом (r). Алгоритм спрямований на пошук у тривимірному параметричному просторі (x , y , r) всіх можливих кіл, які присутні в зображенні.

Зазвичай його використовують для зображення у відтінках сірого, а реалізація перетворення у бібліотеці OpenCV передбачає підбір наступних параметрів:

method - метод детекції, за замовчуванням `cv2.HOUGH_GRADIENT`. Це єдиний метод, підтримуваний в OpenCV, і заснований на градієнтних змінах у зображенні.

dp - відношення роздільної здатності об'ємного простору до зображення. Наприклад, якщо $dp = 1$, роздільна здатність об'ємного простору така ж, як у зображення. Якщо $dp = 2$, роздільна здатність зменшується вдвічі.

MinDist - мінімальна відстань між центрами виявлених кіл. Цей параметр допомагає уникнути накладення кіл, які можуть бути виявлені ближчими одне до одного, ніж значення `minDist`.

param1: Перший параметр використовується для контролю над методом `cv2.Canny()`. Зазвичай це порогове значення для вирізання слабо виражених країв.

param2: Поріг центрування. Менше значення означає, що буде обрано більше хибних кіл, і навпаки.

minRadius: Мінімальний радіус кола, що потрібно виявити.

maxRadius: Максимальний радіус кола, що потрібно виявити.

Приклад детекції вирв із використанням трансформації Хафа наведено на рис. 9

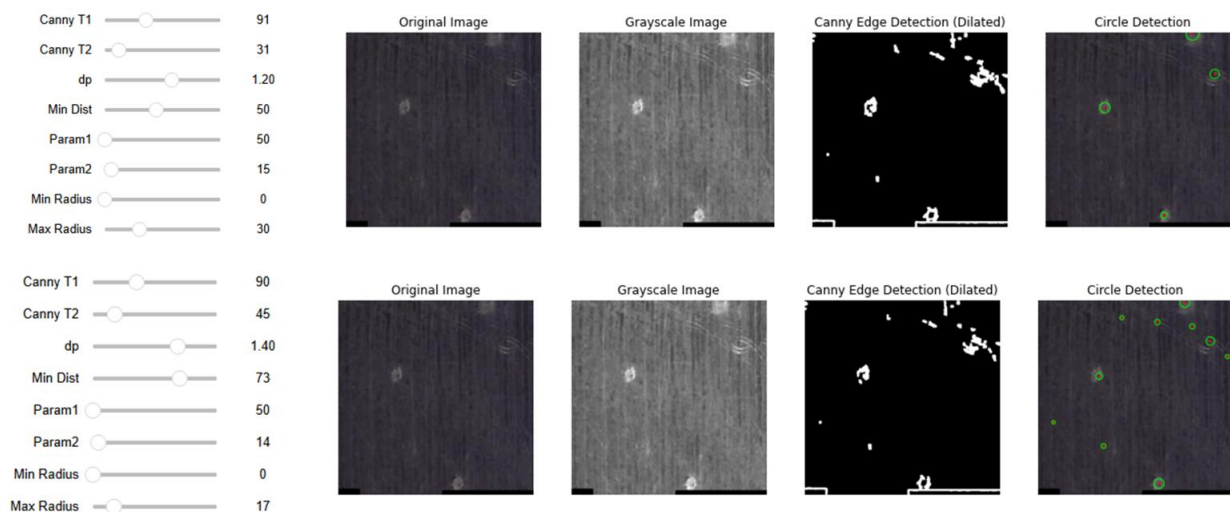


Рис. 9 Приклад детекції вирв із використанням трансформації Хафа.

Як можна побачити із рис. 9, навіть незначна зміна параметрів суттєво впливає на результати детекції вирв, що вимагає постановки та розв'язання задачі підбору гіперпараметрів.

Традиційно оптимізація гіперпараметрів була справою людей, оскільки вони можуть бути дуже ефективними в режимах, коли кількість експериментів дуже обмежена, але сучасні технології та алгоритми роблять можливим проведення більшої кількості експериментів, і алгоритмічні підходи можуть знаходити кращі результати. Наприклад, результати оптимізації гіперпараметрів у задачах навчання нейронних мереж і мереж глибокої віри (DBN). Оптимізація гіперпараметрів із використанням методів випадковий пошуку та або послідовних методів, заснованих на критеріях очікуваного поліпшення дають набагато кращі результати. Випадковий пошук виявився достатньо ефективним для оптимізації гіперпараметрів достатньо складних методів машинного навчання, нейронних мереж. Послідовні алгоритми застосовуються до найскладніших проблем навчання DBN і знаходять значно кращі результати.

Складні моделі машинного навчання такі як мережі глибокої віри (DBN) [2], стеки денойзингових автоенкодерів [3], згорткові мережі [4], а також класифікатори, які ґрунтуються на складних технологіях витягання ознак, можуть мати від десяти до, можливо, п'ятдесяти гіперпараметрів, залежно від того, як експериментатор вирішує параметризувати модель і скільки гіперпараметрів експериментатор вирішує

зафіксувати на розумному за замовчуванням рівні. Таким чином, оптимізацію гіперпараметрів слід вважати формальним зовнішнім циклом у процесі навчання. Навчальний алгоритм, як функція від даних до класифікатора (візьмемо класифікаційні задачі як приклад), включає вибір бюджету щодо того, скільки циклів ЦП має бути витрачено на дослідження гіперпараметрів, і скільки циклів ЦП має бути витрачено на оцінку кожного вибору гіперпараметра (тобто шляхом налаштування звичайних параметрів).

5. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕДУРИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ

Для подальшої оптимізації гіперпараметрів було застосовано бібліотеку Optuna, програмну платформу для автоматичної оптимізації гіперпараметрів, спеціально розроблену для машинного навчання. Вона має імперативний інтерфейс програмування, що базується на принципі "визначити під час виконання". Завдяки розробленому API, що працює за принципом "визначити під час виконання", код, написаний за допомогою Optuna, має високу модульність, а користувач Optuna може динамічно створювати простори пошуку для гіперпараметрів.

При цьому підтримуються різноманітні алгоритми формування вибірки які ефективно звужують простір пошуку, використовуючи записи запропонованих значень параметрів і оцінених значень цілей, що веде до оптимального простору пошуку, який в кінцевому рахунку дозволяє отримати оптимальні значення параметрів.

Серед базових алгоритмів пошуку, що підтримує Optuna варто виділити наступні:

- Градієнтний пошук, реалізований у GridSampler
- Випадковий пошук, реалізований у RandomSampler
- Алгоритм деревоподібної Парзена, реалізований у TPESampler
- Алгоритм на основі CMA-ES, реалізований у CmaEsSampler
- Алгоритм на основі гаусових процесів, реалізований у GPSampler
- Алгоритм для частково фіксованих параметрів, реалізований у PartialFixedSampler
- Алгоритм непомітного сортування генетичних алгоритмів II, реалізований у NSGAIIISampler
- Алгоритм вибірки квазі-Монте-Карло, реалізований у QMCSampler

Процедура пошуку гіперпараметрів виконувалась ітеративно в декілька етапів на кожному з яких відтворювався експеримент, який включав в себе від 100 повторень, які були спрямовані на пошук оптимального значення цільової функції, або досліджуваної метрики (див. рис. 10).

На кожній ітерації метрика розраховувалась виходячи із сформульованих прогнозів та фактичних значень (блок `metric_eval` на рис. 10).

В свою чергу прогнози визначались в середині блоку `make_predictions`, який в залежності від значень досліджуваних гіперпараметрів ідентифікував кола на низці зображень, що використовувались для детекції вив.

Після цього виходячи із координат центрів та радіусів визначених кіл розраховувались відповідні `bounding box`-и, які в свою чергу зіставлялись із наявними даними щодо розмітки зображень. У випадках, коли метрика IoU (Intersection over Union) була більша за 0.5, ідентифікована вирва вважалась коректно ідентифікованою. Всі запропоновані кола, для яких не було відповідних реальних вив відносились до помилково позитивних, а всі реальні вирви на зображені, які не мали відповідних ідентифікованих кіл відносились до помилково-негативних.

Таким чином, в межах кожного повторення опрацьовувались всі наявні зображення і розраховувались базові метрики повноти та точності алгоритму детекції вив.

```
6 def objective(trial):
7     # Define the hyperparameters to optimize
8     param1 = trial.suggest_int('canny_T1', 83, 93)
9     param2 = trial.suggest_int('canny_T2', 32, 60)
10    param3 = trial.suggest_float('dp', 1.1, 1.5)
11    param4 = trial.suggest_int('min_dist', 55, 75)
12    param5 = trial.suggest_int('param1', 45, 47)
13    param6 = trial.suggest_int('param2', 9, 14)
14    param7 = trial.suggest_int('min_radius', 0, 0)
15    param8 = trial.suggest_int('max_radius', 17, 17)
16
17    # Make predictions with the current set of hyperparameters
18    predictions = make_predictions(data,
19        {'canny_T1': param1,
20         'canny_T2': param2,
21         'dp': param3,
22         'min_dist': param4,
23         'param1': param5,
24         'param2': param6,
25         'min_radius': param6,
26         'max_radius': param7
27        })
28
29    # Evaluate the model using your metric function
30    metrics = metric_eval(predictions, gt) # Assuming gt is the ground truth
31    return metrics # Return a metric to maximize
32
33 # Create a study object and optimize
34 study = optuna.create_study(direction='maximize') # 'maximize' for accuracy or other metrics
35 study.optimize(objective, n_trials=100)
```

Рис. 10 Процедура оптимізації гіперпараметрів

На першому етапі експерименту всі параметри досліджувались в довільних широких діапазонах, переважно орієнтуючись на рекомендовані в літературі типові

значення параметрів і експериментально встановлені границі змін, які давали візуально контрольовані зміни в детекції вирв.

За результатами першого етапу експерименту досліджувались розподіл ефективності відносно окремих значень кожного з параметрів (рис 11), а також впливи комбінацій значень параметрів на результати (рис 12).

Для вибору пар параметрів використовували кореляційний аналіз.

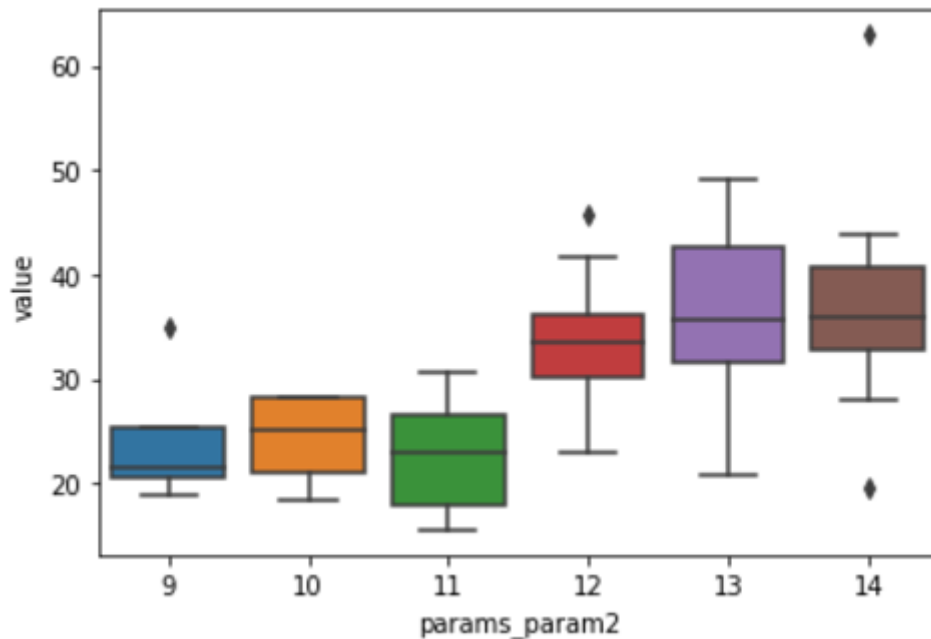


Рис 11. Вплив зміни параметру на значення метрики

При аналізі впливу окремих параметрів на значення метрики особливу увагу треба звертати на верхні значення діапазону (IQR), оскільки саме вони свідчать про те, що відповідне значення параметру має потенціал для покращення цільової метрики.

При дослідженні парних впливів для розділення значень параметрів використовували сигмоїдні перетворення значень метрики, що дозволило більш чітко розділити погані та перспективні діапазони параметрів алгоритму детекції вирв (див. рис. 12)

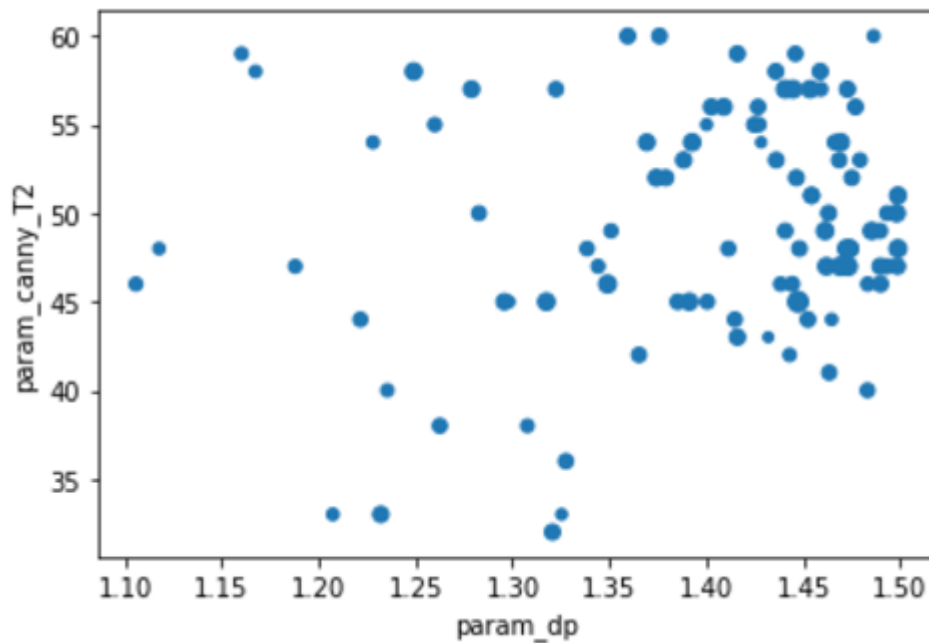


Рис. 12. Дослідження впливу комбінації параметрів на метрику якості алгоритму

Для практичної реалізації запропонованого методичного підходу розроблено концептуальну схему архітектури рішення щодо збору, тренування моделей та обробкою зображень на платформі AWS (рис. 13).

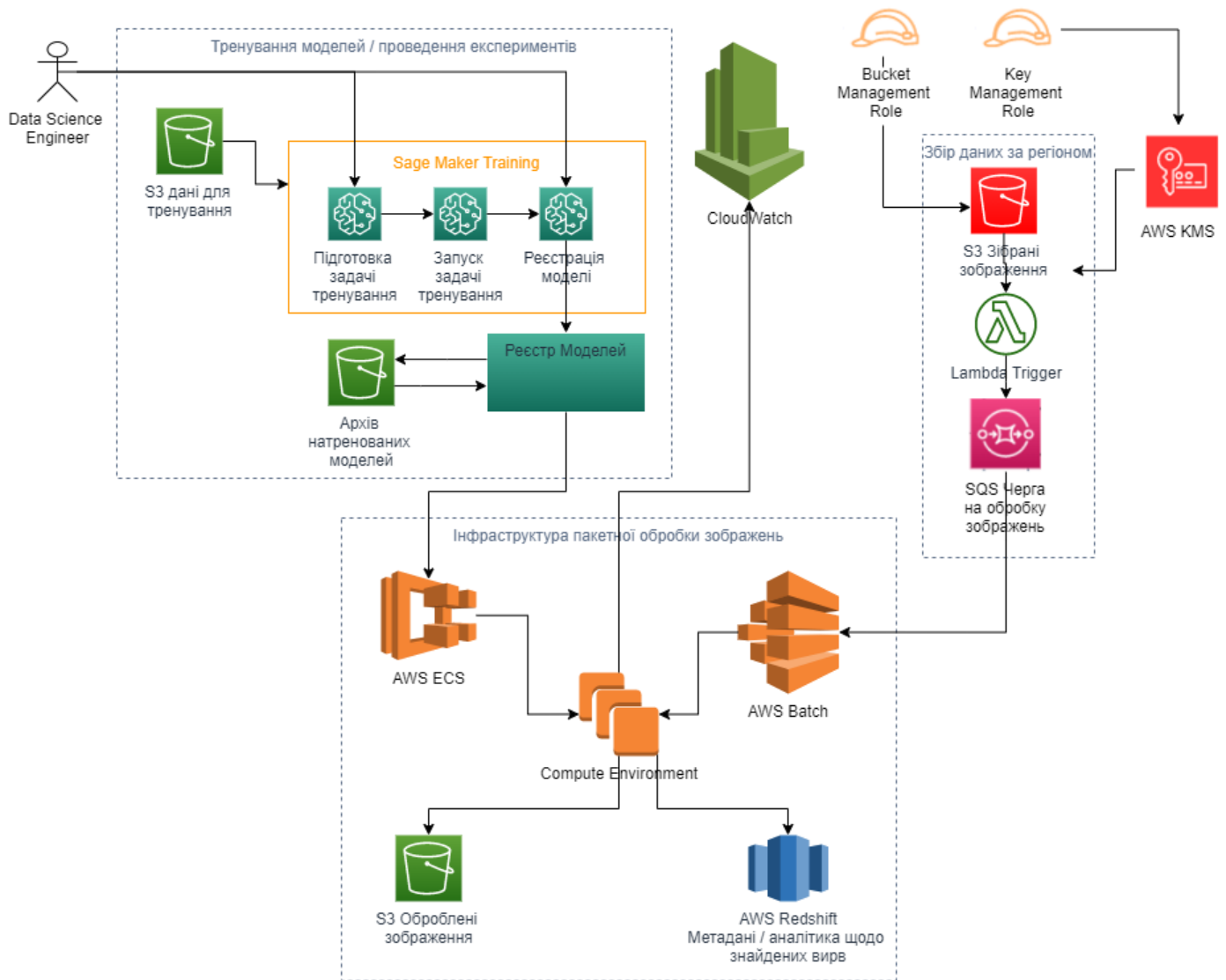


Рис 13. Схема архітектури рішення щодо збору, тренування моделей та обробки зображень на платформі AWS

Ключовими особливостями цієї схеми є забезпечення вибіркового доступу та захисту даних за рахунок використання AWS KMS. Використання KMS для керування ключами, також забезпечує безпеку даних. Застосування сервісів S3 для зберігання та обробки зображень. Для запуску процедур збору та підготовки даних передбачено застосування AWS Lambda, яка використовує тригери для автоматизації відповідних процесів. Тренування, збереження, формування реєстру моделей передбачає використання S3 для зберігання результатів тренування і налаштувань моделей. В цьому компоненті застосування AWS Lambda забезпечує автоматизацію, а також резервне копіювання моделей в Реєстр Моделей. Використання AWS Redshift для аналітики щодо оброблених зображень.

Запропонована архітектура підкреслює інтеграцію різних AWS сервісів для забезпечення ефективного тренування моделей та обробки даних.

6. ОЦІНКА РЕСУРСІВ ДЛЯ РЕКУЛЬТИВАЦІЇ ДЕГРАДОВАНИХ ТЕРИТОРІЙ

Підготовчим етапом до оцінки ресурсів для рекультивації і відповідно компонентом системи є оцінка обсягу ґрунту, необхідного для відновлення деградованих територій. Для цього запропоновано використовувати геометричні формули виходячи із гіпотези, що типовою формою кратеру буде усічений конус.

Для візуалізації масштабів проблеми, доцільним та ефективним є створення комплексної карти деградації земель, яка включає накладання виявлених кратерів на супутникові зображення за допомогою ГІС-інструментів, таких як QGIS, ArcGIS. Базові аспекти подібної роботи окреслені в публікації [3] і можуть стати основою для подальшого дослідження.

Наступний із запропонованих етапів - оцінка ресурсів для рекультивації. Цей етап фокусується на оцінці обсягів ресурсів, необхідних для ефективного відновлення навколишнього середовища. Така оцінка базується на аналізі історичних даних про типи ґрунтів, потреби рослинності та кліматичні умови.

Для забезпечення ефективної реалізації відновлювальних заходів, пропонується розробка детального логістичний плану, який включає в себе задачі із визначення логістичних центрів, транспортування матеріалів, до центрів та розподіл в межах кожного центру. Див. рис. 13.

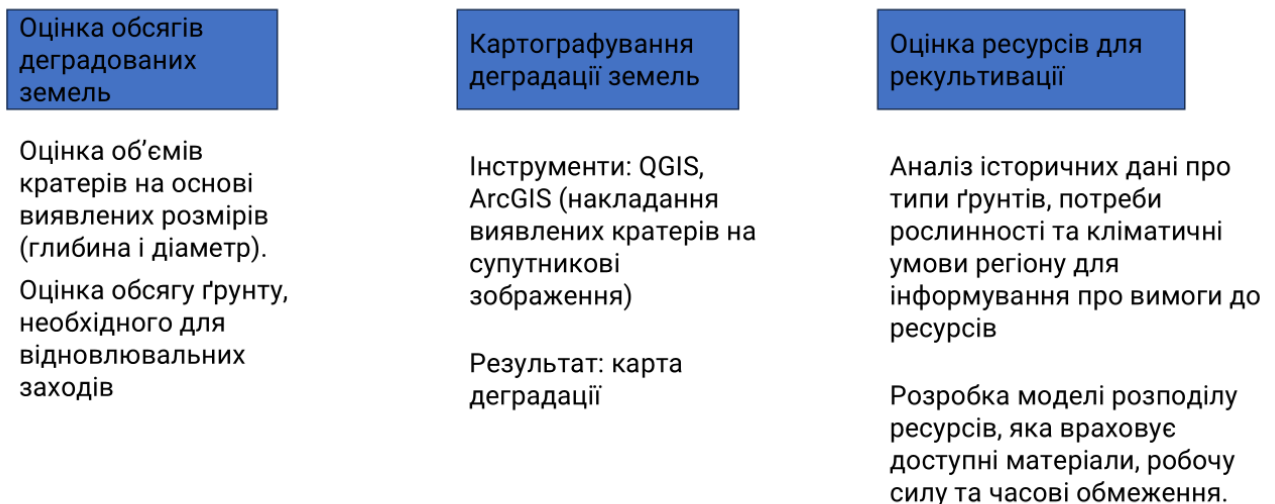


Рис. 13. Етапи оцінки ресурсів для рекультивації деградованих територій

ВИСНОВКИ

Таким чином, впровадження цієї системи дозволяє за рахунок поєднання передових технологій штучного інтелекту, аналізу супутникових знімків та комплексного планування створити потужний інструмент для оцінки та відновлення деградованих земель. Використання запропонованого підходу сприяє відновленню навколишнього середовища та створює основу для більш ефективного управління природними ресурсами у постконфліктних регіонах.

У ході роботи було обґрунтовано необхідність інтеграції сучасних технологій, таких як трансфертне навчання на основі моделі PyCDA, для адаптації алгоритмів до специфічних умов пошкоджених територій. Це дозволяє суттєво знизити витрати на створення нових моделей, забезпечуючи при цьому високу точність і універсальність результатів.

Розроблена методика навчання моделей комп'ютерного зору базується на ретельній підготовці зібраних і розмічених даних. Це включає попередню обробку, аугментацію та валідацію датасетів, що значно підвищує ефективність алгоритмів, таких як згорткові нейронні мережі, у завданнях детекції та оцінки пошкоджень.

Особливості використання евристичних алгоритмів для детекції контурів забезпечили можливість автоматизації процесу ідентифікації вирв навіть у випадках складних геометричних форм. Виявлення об'єктів за допомогою перетворення Хафа дозволило досягти високої точності при оцінці масштабів пошкоджень.

Оптимізація гіперпараметрів із застосуванням бібліотеки Optuna підтвердила свою ефективність у підвищенні продуктивності алгоритмів машинного навчання. Такий підхід дозволяє досягти збалансованості між точністю і швидкістю обробки даних, що є критичним для систем оперативного моніторингу.

Оцінка ресурсів для рекультивації деградованих територій засвідчила доцільність застосування геометричних моделей (усіченого конуса) для розрахунку обсягів ґрунту та інших матеріалів. Візуалізація результатів за допомогою ГІС-технологій дозволяє створювати детальні карти пошкоджень, які слугують основою для прийняття управлінських рішень.

Запропонована система інтегрує наукові підходи та практичні рішення, що забезпечує її застосування в умовах постійних змін рельєфу, спричинених військовими діями. Вона також створює умови для подальшого розвитку технологій екологічного моніторингу та управління земельними ресурсами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Benedix, G. K., Norman, C. J., Bland, P. A., Towner, M. C., Paxman, J., & Tan, T. (2018). Automated detection of Martian craters using a convolutional neural network. *49th Lunar and Planetary Science Conference*. Retrieved from <https://www.hou.usra.edu/meetings/lpsc2018/pdf/2202.pdf>
2. Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Wu, K. (2019). Lunar crater identification via deep learning. *Icarus*, 317, 27-38. <https://doi.org/10.1016/j.icarus.2018.06.022>
3. Weir, D., & McQuillan, D. (2019). Mapping military impacts on the environment: A GIS-based approach to conflict pollution. *Journal of Political Ecology*, 26(1), 697-718. <https://doi.org/10.2458/v26i1.23169>
4. Yao, J., et al. (2018). "Adaptive Hough Transform for Circle Detection." *IEEE Transactions on Image Processing*, 27(6), 2720-2733.
5. Zhang, L., et al. (2016). "Robust Circle Detection via Parameter Optimization." *Pattern Recognition*, 54, 35-48.
6. Akiba, T., et al. (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623-2631.
7. Shen, L., et al. (2021). "Efficient Hyperparameter Optimization Techniques in Machine Learning." *Neural Computing and Applications*, 33(12), 7381-7396.
8. Kussul, N., Drozd, S., Yailymova, H., Shelestov, A., Lemoine, G., & Deininger, K. (2023). Assessing damage to agricultural fields from military actions in Ukraine: An integrated approach using statistical indicators and machine learning. *Remote Sensing Applications: Society and Environment*, 31, 100976. <https://doi.org/10.1016/j.rsase.2023.100976>
9. Canny, J. (1986). "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
10. Ziou, D., & Tabbone, S. (1998). "Edge Detection Techniques: A Review." *International Journal of Pattern Recognition and Image Analysis*, 8(4), 537-559.
11. YOLOv5 Architecture: Jocher, G., et al. (2020). "YOLOv5 Repository Documentation." GitHub, ultralytics/yolov5.
12. Wang, C., et al. (2020). "YOLOv5: Improved Object Detection Architecture." *arXiv preprint arXiv:2004.10934*.

ДОДАТКИ

Додаток А. Регіон для збору даних



ДОДАТОК Б. Приклади зображення рівнів наближення
ZOOM 17

ZOOM 18
ZOOM 19