

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ ОБ'ЄКТІВ НА
НЕВЕЛИКІЙ ГЛИБИНІ З ВИКОРИСТАННЯМ ЕВОЛЮЦІЙНИХ
МЕТОДІВ**

(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-3

Машков В. В.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Шафроненко А.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____
(підпис)

Кобилін О. А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві

Машкову В'ячеславу Вадимовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебзастосунку для пошуку об'єктів на невеликій глибині з використанням еволюційних методів

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 2 червня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV, фреймворк Flask, бібліотека React, мова програмування Python, мова JavaScript, бібліотеки NumPy та SciPy для обробки даних, Matplotlib для візуалізації, бібліотека scikit-image для обробки зображень, алгоритми генетичної оптимізації.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд методів детектування об'єктів на сканах георадару.2. Методи попередньої обробки сканів георадару.3. Використання еволюційних алгоритмів для пошуку об'єктів на радарограмах.4. Створення вебзастосунку для аналізу сканів георадару.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми пошуку об'єктів на невеликій глибині, постановка задачі, тестові зображення, опис алгоритму, огляд застосунку.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

доц. Шафроненко А.Ю
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 69 с., 33 рис., 30 джерел.

ГЕНЕТИЧНИЙ АЛГОРИТМ, ПЕРЕТВОРЕННЯ ХАФА, ДЕТЕКТУВАННЯ КРИВИХ, ВІДРІЗКІВ, ІНТЕГРАЛЬНЕ ПЕРЕТВОРЕННЯ, ГЕОРАДАР, ОБРОБКА СИГНАЛІВ.

Об'єктом роботи є радарограми, що містять відображення підповерхневих об'єктів у вигляді параболічних кривих.

Метою роботи є розробка методів, що базуються на використанні генетичних алгоритмів, для автоматичного знаходження параболічних кривих на георадарних сканах, з подальшою реалізацією функціоналу у вигляді інтерактивного вебзастосування.

Використано методи попередньої обробки сигналів та генетичні алгоритми. Проведено дослідження методів пошуку об'єктів на даних з георадару, розроблено алгоритм пошуку об'єктів на даних з георадару.

У результаті роботи здійснена програмна реалізація системи для пошуку об'єктів на невеликій глибині за допомогою георадару.

GENETIC ALGORITHM, TRANSFORM HOUGH, CURVE DETECTION, SEGMENT DETECTION, INTEGRAL TRANSFORMATION, GROUND PENETRATION RADAR, SIGNAL PROCESSING.

The object of the work is radargrams containing the display of subsurface objects in the form of parabolic curves.

The purpose of the work is to develop methods based on the use of genetic algorithms for automatic detection of parabolic curves on GPR scans, with the subsequent implementation of the functionality in the form of an interactive web application.

Signal preprocessing methods and genetic algorithms were used. Methods for searching for objects on GPR data were studied, and an algorithm for searching for objects on GPR data was developed.

As a result of the work, the software implementation of a system for searching for objects at shallow depths using GPR was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Актуальність проблеми	10
1.1.1 Стан області дослідження	10
1.1.2 Роль автоматизації інтелектуального аналізу даних	11
1.2 Огляд технологій та методів	13
1.2.1 Основи радарних вимірювань.....	13
1.2.2 Методи обробки GPR-зображень	14
1.2.3 Існуючі рішення	16
1.3 Постановка задачі	19
2 Опис алгоритму виявлення параболічних відбиттів	21
2.1 Формалізація задачі	21
2.2 Попередня обробка даних з георадару	22
2.2.1 Віднімання фонового сигналу	22
2.2.2 Нормалізація та підсилення контрасту	25
2.2.3 Фільтрація шуму	26
2.2.4 Кадрування області інтересу.....	27
2.3 Налаштування алгоритму згідно з умовами задачі	29
2.3.1 Пояснення вибору генетичного алгоритму	29
2.3.2 Основні поняття генетичного алгоритму	30
2.3.3 Функція пристосованості	31
2.3.4 Штраф за малу кривизну параболи	32
2.3.5 Відбір.....	33
2.3.6 Схрещування	34
2.3.7 Мутація.....	34
2.3.8 Елітизм	35
2.3.9 Параметри алгоритму	35

	6
2.3.10 Адаптація алгоритму для пошуку декількох об'єктів на радарограмі.....	36
2.3.11 Фінальний вигляд алгоритму	37
3 Програмна реалізація вебзастосунку	39
3.1 Вибір середовища програмної реалізації	39
3.2 Програмна реалізація.....	42
3.2.1 Створення бекенду.....	42
3.2.2 Попередня обробка зображення.....	51
3.2.3 Створення алгоритму.....	51
3.2.4 Створення фронтенду.....	55
Висновки	65
Перелік джерел посилання	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GPR – Ground Penetration Radar (георадар)

SNR – Signal-to-Noise Ratio (співвідношення сигнал/шум)

FDTD – Finite-Difference Time-Domain (метод скінченних різниць у часовій області)

CLAHE – Contrast Limited Adaptive Histogram Equalization (обмежене адаптивне вирівнювання гістограми)

GA – Genetic Algorithm (генетичний алгоритм)

ES – Evolution Strategy (еволюційна стратегія)

DE – Differential Evolution (диференційна еволюція)

CMA-ES – Covariance Matrix Adaptation Evolution Strategy (еволюційна стратегія з адаптацією матриці коваріації)

HTTP – HyperText Transfer Protocol (протокол передавання гіпертексту)

API – Application Programming Interface (інтерфейс прикладного програмування)

IDE – Integrated Development Environment (інтегроване середовище розробки)

REST – Representational State Transfer (передача репрезентативного стану)

ВСТУП

Георадарне зондування підземних шарів ґрунту стало невід'ємною частиною сучасної геології, археології та будівництва та інших галузей. Одним із засобів для дослідження структури ґрунтів без їх фізичного руйнування є георадар (GPR). Принцип його дії ґрунтується на реєстрації електромагнітних хвиль, відбитих від підземних об'єктів, неоднорідностей чи шарів. Завдяки своїй високій просторовій роздільній здатності та здатності виявляти об'єкти на різних глибинах, георадар набув широкого поширення.

GPR активно використовується в усьому світі для дослідження фундаментів історичних споруд, визначення підземних інженерних комунікацій, а також оцінці стану дорожнього покриття. У нас час найбільш перспективним є використання георадару у гуманітарному розмінванні. Популярність даної технології спричинила зростання обсягів отримуваних даних, що вимагає вдосконалення методів їх обробки.

В той же час інтерпретація отриманих з георадару даних все ще залишається складним завданням. Зображення, сформовані в результаті зондування, часто є зашумленими, містять артефакти, сигнали від нецільових об'єктів або неоднорідностей середовища. Крім того, підземні об'єкти можуть мати складну форму, що ускладнює їх розпізнавання. Аналіз радарограм вимагає високої кваліфікації оператора, великого досвіду роботи з GPR-даними та значного часу. У реальних умовах, коли прийняття рішень повинно бути оперативним, ручна обробка не завжди є ефективною.

У зв'язку з цим актуальним є пошук підходів, що дозволяють автоматизувати обробку георадарних зображень. Одним із перспективних напрямів є застосування еволюційних алгоритмів, які засновані на принципах природного відбору. Вони здатні ефективно працювати в умовах високої шумності, фрагментарності даних та багатовимірних просторах пошуку, що робить їх доцільними для задач виявлення характерних структур у георадарних зображеннях.

Параболічні відбиття на GPR-зображеннях формуються при фокусному відбитті хвиль від компактного об'єкта на певній глибині. В контексті гуманітарного розмінування, такі відбиття можуть свідчити про наявність вибухових пристроїв, особливо таких, що не мають металу в своїй конструкції, і тому не можуть бути знайдені звичайним металошукачем. Тому задача автоматичного виявлення таких структур на радарограмі має практичну цінність.

Еволюційні алгоритми ж ефективними у задачах з високим рівнем зашумленості даних, також у задачах класифікації цільових сигналів та сегментації зображень. Також успішно використовуються підходи до автоматичного налаштування фільтрів або пошуку криволінійних об'єктів на радарограмах.

Актуальність роботи полягає у зростаючій потребі в ефективних та автоматизованих методах аналізу георадарних даних для виявлення об'єктів на невеликій глибині. Традиційна інтерпретація радарограм вимагає значного досвіду та часу, що обмежує швидкість та точність прийняття рішень у сферах, де оперативність є критичною.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

1.1.1 Стан області дослідження

Георадар – це прилад для зондування ґрунту за допомогою електромагнітних імпульсів високої частоти. Його робота базується на принципі випромінювання коротких імпульсів у діапазоні від десятків мегагерц до кількох гігагерц, які проникають у ґрунт та інші матеріали. Коли ці хвилі досягають межі розділу середовищ із різною діелектричною проникністю (наприклад, між сухим ґрунтом і металевим об'єктом), частина сигналу відбивається назад і фіксується приймальною антеною. Час запізнення між випромінюванням імпульсу та отриманням відбитого сигналу дозволяє оцінити глибину, на якій знаходиться об'єкт, а амплітуда відбиття – зробити висновки про матеріальні властивості середовища [1–6].

Форма та інтенсивність відбитих сигналів залежать від типу ґрунту, вологості, наявності неоднорідностей, глибини залягання об'єктів, кута їх нахилу та розміру. Це дозволяє створювати зображення структури підповерхневого середовища з високою точністю.

Вимірювання георадару можна представити як двовимірне зображення, на якому відображаються сигнали, отримані при скануванні вздовж певного профілю. Об'єкти, що мають малу площу в поперечному перерізі (наприклад, труби або камені), як правило, формують на таких зображеннях характерні параболічні відбиття. Це пов'язано з тим, що коли антена георадару переміщується над об'єктом, сигнал спершу долає більшу відстань (по дотичній), а з наближенням до центру відбивається швидше. Таким чином, відгук набуває параболічної форми з вершиною, розташованою безпосередньо над об'єктом.

Георадіолокаційні методи характеризуються відсутністю механічного впливу на досліджувану поверхню, що дозволяє використовувати їх у

ситуаціях у ситуаціях з обмеженим доступом до об'єкта дослідження. Наприклад, на будівництві часто необхідно розуміти розташування підземних комунікацій, щоб не пошкодити їх, але там не завжди є інформація про їхню точну локацію. За допомогою георадару можна проаналізувати шар ґрунту та визначити небезпечні ділянки. Також георадар використовують для виявлення археологічних артефактів або старовинних поховань. Також це дозволяє аналізувати потенційно заміновані ділянки на предмет пластикових або інших вибухових пристроїв, які не можна знайти металодетектором [2].

В той же час у георадару є і свої мінуси. Якість та глибина проникнення сигналу залежать від властивостей середовища, наприклад, у сухому піску або льоді сигнал проходить глибше, ніж у вологому ґрунті або глині. Також, результати сканування потребують інтерпретації, що є достатньо складним завданням і вимагає спеціальної кваліфікації оператора радару.

1.1.2 Роль автоматизації інтелектуального аналізу даних

При зондуванні значної території генеруються великі обсяги даних, які складно обробити вручну. Ручна інтерпретація такої кількості даних займає багато часу і є вразливою до суб'єктивності оцінки [7–10]. Тобто різні спеціалісти можуть по-різному трактувати ті самі відбиття на радарограмі у випадку зашумлених або неоднорідних даних. Також тут присутній фактор концентрації спеціаліста, коли він переглядає достатньо велику кількість сканів, або навіть дані з декількох різних радарів, деякі відбиття можуть бути пропущені [6]. Через це автоматизація процесів аналізу за допомогою інтелектуальних алгоритмів є дуже важливою при роботі з GPR-даними.

Використання методів інтелектуального аналізу дає змогу розв'язувати складні задачі, пов'язані з пошуком характерних відбиттів у сигналі, виділенням закономірностей у шумових даних та оптимізацією параметрів моделей для опису фізичних явищ [11–12]. При виявленні підземних об'єктів

часто йдеться не про чіткі геометричні контури, а про складні просторово-часові залежності, які добре піддаються аналізу за допомогою алгоритмів оптимізації [13].

Для виявлення таких залежностей існують стандартні методи обробки георадарних зображень, однак в деяких випадках доцільно використовувати еволюційні алгоритми, наприклад генетичний алгоритм (GA). Перевага такого типу алгоритмів полягає в здатності знаходити глобальні оптимуми у задачах з великою кількістю локальних екстремумів – таку природу мають функції помилки при пошуку підземних об'єктів у GPR-зображеннях. На відміну від класичних методів, які можуть застрягати у локальному максимумі, еволюційні підходи проводять пошук у всьому просторі можливих рішень, комбінуючи та змінюючи параметри, щоб поступово вдосконалювати популяцію гіпотетичних рішень [14]. Така особливість робить ці методи достатньо корисними у різних типах задач.

Для пошуку параболічних відбиттів – характерних ознак для точкових об'єктів – генетичний алгоритм може бути застосований для автоматичного підбору параметрів параболи, які найкраще узгоджуються з відбиттям у сигналі. Це означає, що модельна парабола «еволюціонує» в процесі оптимізації, поступово наближаючись до реального відбиття, навіть якщо воно частково спотворене або знаходиться на зашумленому фоні. Однак якість роботи алгоритму залежить від правильного задання функції пристосованості, а також від якості видалення шумів.

Окрім безпосереднього виявлення відбиттів, еволюційні алгоритми можуть застосовуватись також для налаштування гіперпараметрів попередньої обробки сигналу – наприклад, вибору типу та ширини фільтра, глибини порогового відсікання, або нормалізації сигналу. Це дозволяє формувати адаптивні системи обробки, які самостійно підлаштовуються до характеристик ґрунту, частоти сигналу, рівня шуму, що значно покращує результати обробки без втручання оператора. Така особливість є важливою для гуманітарного розмінування.

1.2 Огляд технологій та методів

1.2.1 Основи радарних вимірювань

Фізичні принципи проходження електромагнітної хвилі в ґрунті базуються діелектричній проникності, провідності і магнітній проникності середовища. Коли радіохвильовий імпульс, створений передавачем георадару, потрапляє в ґрунт, швидкість його поширення та ступінь загасання визначаються комплексною діелектричною константою середовища. Верхні шари ґрунту часто містять вологу, частинки глини та органічні речовини, що призводить до високого електричного опору та значного поглинання енергії хвилі. З іншого боку, щільний сухий пісок або скельні породи пропускають сигнал з меншими втратами, але можуть спричинити багаторазові відбиття на межах неоднорідностей. Крім того, неоднорідності розподілу вологи та структурні фракції розсіюють хвилі, що призводить до додаткового шуму в отриманих даних.

Формування імпульсу починається в генераторі георадару, де високочастотний коливальний контур виробляє короткий електромагнітний спалах з тривалістю від десятків наносекунд до кількох сотень наносекунд. Цей імпульс через антену випромінюється в ґрунт, проникає на певну глибину та взаємодіє з неоднорідностями середовища. Після цього відбитий сигнал приймається антеною-детектором і підсилюється у приймальному тракті. Далі за допомогою аналого-цифрового перетворювача імпульси оцифровуються, формуючи послідовність відліків інтенсивності ехо-сигналу в залежності від часу затримки [1–3, 6].

Малий об'єкт, наприклад закопаний камінь або труба, розсіює частину енергії імпульсу назад до антени. Через кутові відмінності прямих і відбитих шляхів час приходу сигналу змінюється таким чином, що на радіографії виявляється крива у формі параболи. Відсутність ідеальної однорідності середовища та шумові характеристики системи можуть дещо спотворювати

форму, але в основному відбиття від точкового чи лінійного об'єкта має вигляд параболи [6].

1.2.2 Методи обробки GPR-зображень

Шум і перешкоди в GPR-зображеннях суттєво погіршують якість даних і можуть призводити до хибних інтерпретацій параболічних відбитків. Достатньо поширеною проблемою є *multipath propagation*, коли електромагнітні хвилі проходять декілька шляхів від антени до об'єкта та назад через різні шари середовища. Такі подвійні або кратні шляхи подовжують час приходу сигналу та створюють додаткові відбиття, що зміщують або дублюють справжні параболи на радіографі. Другим джерелом перешкод є відлуння (*ringing*) в антенній системі – затримані зворотні коливання в антені та фільтрах залишаються в сигналі після проходження основного імпульсу, формуючи додаткові осциляції довкола справжніх відбиттів. Ці відлуння можуть маскувати слабкі параболи або створювати похибки в оцінці їхніх параметрів. Крім того, неоднорідності середовища (глибинні шари з різною вологістю, наявність кореневої системи рослинності чи дрібних включень) призводять до розсіювання та дифракції хвиль, що значно ускладнює виділення цільових відбиттів на фоні шумів різної інтенсивності. Всі ці фактори знижують сигнал-шумове відношення (SNR) і тому виникає необхідність попередньої обробки даних.

Для зниження впливу шумів та перешкод на початковому етапі аналізу GPR-зображень застосовують різноманітні лінійні та спектральні фільтри. Лінійні фільтри працюють безпосередньо в часовій (або просторій) області й реалізуються згортокою сигналу з відповідним ядром. До найпростіших належать ковзні середні (*moving average*) та гауссівські фільтри, які пригладжують високочастотні коливання, що часто відповідають випадковим шумам та відлунням. Для прибирання повільних коливань фону

використовують фільтри віднімання фону (background removal), коли від кожного профілю віднімається середній сигнал за обраний набір сусідніх профілів. Спектральні фільтри, натомість, оперують перетворенням Фур'є: сигнал переводять у частотну область, де застосовують високочастотні та низькочастотні (band-pass) фільтри для виокремлення корисного діапазону частот, характерного для параболічних відбиттів, і придушення вузькосмугових перешкод або сплесків низькочастотного фону. Часто також використовують notch-фільтри для вилучення періодичних інтерференцій, що виникають через електромагнітні джерела поблизу. Після обробки в частотній області відфільтрований спектр переводять зворотним перетворенням Фур'є, отримуючи очищену радарограму. Комбінована послідовність лінійних та спектральних кроків значно покращує SNR і формує базу для подальшого застосування алгоритмів виявлення парабол [2].

Класичні методи виявлення об'єктів на зображеннях, ґрунтуються на обробці зображень з метою виокремлення характерних геометричних структур. Ці методи ефективно працюють у випадках, коли форма шуканого об'єкта відома наперед, а зображення має достатньо високе сигнал-шумове співвідношення [11].

Одним з поширених методів є Hough-перетворення (Hough Transform) – алгоритм, який дозволяє виявляти геометричні криві заданого типу (прямі, кола, параболи тощо) на бінаризованому зображенні. Його принцип полягає у проєкції кожної точки зображення в параметричний простір можливих кривих. Наприклад, для прямої кожна точка (x, y) відповідає безлічі прямих, які проходять через неї, кожна з яких описується параметрами (ρ, θ) . Для параболи рівняння має форму $y = ax^2 + bx + c$, тож кожна точка на зображенні задає поверхню в тривимірному параметричному просторі (a, b, c) . Алгоритм «голосує» за всі можливі криві, які можуть проходити через кожну точку, і в результаті акумулює кількість голосів у тривимірному масиві. Ті області простору параметрів, які отримали найбільше голосів, відповідають ймовірному розташуванню параболи на зображенні. Основним недоліком

Hough-перетворення для парабол є велика обчислювальна складність, оскільки зростає кількість параметрів – на відміну від ліній, для яких простір двовимірний, у випадку парабол потрібно оцінювати три або більше параметрів, а також враховувати масштаб і поворот, якщо об'єкти мають варіативну орієнтацію (у випадку коли радар рухається по нерівній поверхні).

Іншим підходом є застосування детекторів контурів, які дозволяють спочатку виділити межі яскравих об'єктів, а потім аналізувати їхню форму. До найвідоміших належать оператори Собеля, Робертса, Прюїта, а також більш складні варіанти, як-от детектор Кенні (Canny edge detector). Алгоритм Кенні працює в кілька етапів: згладжування зображення гауссівським фільтром, обчислення градієнтів, пошук локальних максимумів, придушення неконтурних пікселів та двопорогова фільтрація для відбору найбільш релевантних країв. У результаті формується набір пікселів, які утворюють контур об'єкта. Ці контури можна далі аналізувати шляхом апроксимації – наприклад, методом найменших квадратів підібрати параболу до знайденого краю. Але за наявності шумів контур може виявитися розірваним або мати зайві вигини, що потребує попереднього згладжування, морфологічної обробки або фільтрації коротких контурів.

Класичні методи виявлення мають переваги у швидкодії (особливо при апаратному прискоренні) та добре працюють на синтетичних або слабо зашумлених даних. Проте вони втрачають ефективність при складних геометричних формах, перешкодах і неоднорідному фоні [11, 15].

1.2.3 Існуючі рішення

Існує декілька рішень для роботи з GPR даними. GprMax – симулятор на основі методу скінченних різниць у часі (FDTD) дозволяє моделювати проходження електромагнітних хвиль через шари середовища довільної геометрії. GprMax написаний на Python і C++, має модульну архітектуру для

підключення різних типів антен і середовищ, підтримує одновимірні, двовимірні та тривимірні моделі, а також формати вводу-виводу, сумісні з MATLAB та ParaView. GprMax має відкритий код, можливість деталізованого налаштування фізичних параметрів середовища (діелектрична проникність, питомий опір, неоднорідності) та може відтворювати результати у різних експериментальних сценаріях. Однак GprMax не має функції автоматичної обробки радарограм.

Інший доступний продукт це Ekko_Project від компанії Sensors & Software. Він пропонує широкий функціонал, від попередньої обробки (фільтрація, компенсація середовища, фонові корекції) до тривимірної візуалізації, глибинного профілювання та створення інтерактивних хмар точок. В Ekko_Project реалізовані модулі автоматичного розпізнавання лінійних та дугоподібних відбиттів, серійна обробка даних зі стаціонарних обстежень та експорт результатів у популярні формати GIS. Але він є достатньо дорогим, та він не має способів розширення або модифікації функціоналу. Ekko_Project добре справляється зі стандартними профілями, але для виявлення слабких або глибоко залягаючих об'єктів він потребує налаштування та постійної присутності оператора.

Програмний комплекс RADAN виробництва компанії Geophysical Survey Systems Inc. (GSSI). Функціональні можливості системи охоплюють повний цикл аналізу радіолокаційної інформації, включаючи попередню обробку сигналів, фільтрацію шумів та візуалізацію результатів. Система дає можливість використовувати алгоритми цифрової обробки сигналів для підвищення співвідношення корисного сигналу та шуму, також програма має декілька методів детекції гіпербол, що спираються на попередню обробку, наприклад кореляційний аналіз, частотно-часовий аналіз, та аналіз амплітудних характеристик. Але методи детекції, що наявні в програмі, характеризуються підвищеною чутливістю до фонових шумів та геологічних неоднорідностей, до того ж кореляційний аналіз (шаблонний підхід) не може підбирати параметри динамічно, а лише порівняти з еталоном, що не завжди є

ефективним в реальних умовах. В той же час програма дозволяє автоматизувати процеси за допомогою .NET API, що може бути дійсно корисним при її інтеграції в більш загальну систему аналізу даних з радарів.

Застосунок REFLEXW, розроблений компанією Sandmeier Scientific Software, представляє рішення для обробки та інтерпретації георадарних даних з розширеними можливостями геофізичного моделювання. Архітектура системи базується на модульному принципі організації функціональних блоків, що забезпечує гнучкість налаштування робочого процесу відповідно до специфіки конкретних геофізичних завдань. Програма підтримує широкий спектр форматів вхідних даних від різних виробників георадарного обладнання та забезпечує інтеграцію з геоінформаційними системами для просторового аналізу результатів досліджень.

REFLEXW реалізує декілька підходів до автоматизованого пошуку параболічних відбиттів, включаючи модифіковані методи шаблонного розпізнавання з адаптивними масками. Система використовує алгоритм Хафа для детекції гіперболічних кривих у просторі параметрів, що дозволяє ідентифікувати об'єкти навіть при частковому перекритті їх сигнатур фоновими завадами. Додатково програма включає модуль кластерного аналізу для групування виявлених аномалій за геометричними та амплітудними характеристиками.

Інтегровані алгоритми спектрального аналізу забезпечують виділення характерних частотних компонентів, притаманних дифракційним відбиттям від локальних неоднорідностей. Система застосовує банки цифрових фільтрів з адаптивними параметрами для підвищення контрастності гіперболічних структур відносно геологічного фону. Використання вейвлет-перетворень дозволяє здійснювати локалізований частотно-часовий аналіз сигналу з метою виявлення слабо виражених аномалій.

Незважаючи на наявність адаптивних масок, система REFLEXW потребує значного обсягу попереднього налаштування параметрів детекції під конкретні умови зйомки. Оператор має вручну встановлювати діапазони

швидкостей електромагнітних хвиль, геометричні обмеження пошуку та порогові значення амплітудних критеріїв. Це знижує рівень автоматизації процесу та потребує високої кваліфікації персоналу для ефективного використання системи.

Традиційні методи, реалізовані в REFLEXW, базуються на статичних алгоритмах з фіксованою логікою прийняття рішень. Відсутність механізмів самоадаптації не дозволяє системі автоматично пристосовуватися до мінливих умов сигнально-завадової обстановки та специфічних характеристик досліджуваних об'єктів.

1.3 Постановка задачі

Таким чином, виявлення параболічних відбиттів на георадарних зображеннях є актуальним завданням для автоматизованого пошуку невеликих об'єктів під землею. Тому ставиться завдання розробки алгоритму пошуку геометричних структур (парабол), який поєднує попередню обробку радарограм із застосуванням генетичного алгоритму для точного знаходження кривих.

Об'єктом роботи є радарограми, що містять відображення підповерхневих об'єктів у вигляді параболічних кривих.

Метою роботи є розробка методів, що базуються на використанні генетичних алгоритмів, для автоматичного знаходження параболічних кривих на георадарних сканах, з подальшою реалізацією функціоналу у вигляді інтерактивного вебзастосунку.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів обробки георадарних зображень та виявлення геометричних структур;
- дослідити застосування еволюційних методів для знаходження параметричних кривих на зображеннях;

- розробити алгоритм пошуку парабол на основі генетичного алгоритму;
- реалізувати модуль попередньої обробки зображення для підвищення точності детекції;
- створити комп'ютерну модель та реалізувати вебзастосунок для виявлення параболічних відбиттів;
- протестувати розроблений інструмент на прикладних георадарних даних.

2 ОПИС АЛГОРИТМУ ВИЯВЛЕННЯ ПАРАБОЛІЧНИХ ВІДБИТТІВ

2.1 Формалізація задачі

Задача виявлення параболічних сигналів у даних GPR полягає в автоматичному знаходженні на двовимірному зображенні (В-скані) аркових кривих, які утворюються відбитими сигналами від підземних об'єктів. Кожен такий об'єкт (металева труба, пустота, коріння дерев тощо) на виході формує послідовність імпульсів, що поглинаються й відбиваються під певним кутом, і на В-скані ці імпульси спостерігаються у вигляді парабол.

Нехай вхідними даними є матриця інтенсивностей $I(i,j)$, де i – індекс глибини (рядок), а j – індекс пройденої відстані уздовж профілю (стовпець). Значення $I(i,j)$ відображає амплітуду прийнятого сигналу в точці (i,j) . Параболічний сигнал в ідеалізованому випадку описується рівнянням:

$$i = a(j - x_0)^2 + y_0. \quad (2.1)$$

Отже, виявлення об'єкта зводиться до знаходження трійки параметрів, що максимально узгоджуються з інтенсивностями пікселів у зображенні:

$$\theta = (x_0, y_0, a), \quad (2.2)$$

де x_0 – горизонтальна координата вершини параболи (положення об'єкта вздовж профілю);

y_0 – вертикальна координата вершини (глибина залягання);

a – параметр кривизни параболи.

Формально, потрібно вирішити задачу оптимізації:

$$\theta^* = \arg \max_{\theta} F(\theta; I). \quad (2.3)$$

При цьому треба враховувати, що:

- у реальних GPR-даних присутній шум, неоднорідності ґрунту та перешкоди від інших відбиттів, тому значення $I(i,j)$ можуть містити як корисні, так і хибні піки;
- кількість і положення об'єктів із параболічними відбитками невідомі заздалегідь;
- може бути від однієї до декількох кривих, що іноді накладаються;
- параметри θ обмежені в діапазонах:

$$x_0 \in [0, W), \quad y_0 \in [0, H), \quad a \in [a_{min}, a_{max}]. \quad (2.4)$$

2.2 Попередня обробка даних з георадару

2.2.1 Віднімання фонового сигналу

Після отримання вимірювань з георадару необхідно провести попередню обробку зображення. Основним етапом є усунення горизонтального фону, що виникає внаслідок потужного відбиття сигналу від самого пристрою, поверхні ґрунту або від інших однорідних горизонтальних шарів. Такі компоненти мають вигляд суцільних ліній на радарограмі й суттєво ускладнюють автоматичне виявлення об'єктів. Крім того, горизонтальний фон має тенденцію перекривати або приглушувати більш слабкі сигнали від об'єктів, що знаходяться глибше, а також формує сильну низькочастотну складову, яка негативно впливає на якість функції пристосованості при подальшому використанні еволюційного алгоритму.

Приклад вимірювань з георадару без попередньої обробки показано на рисунку 1.1.

Щоб зменшити вплив горизонтальних компонент, необхідно використати декілька підходів для фільтрації фону [6, 16–17].

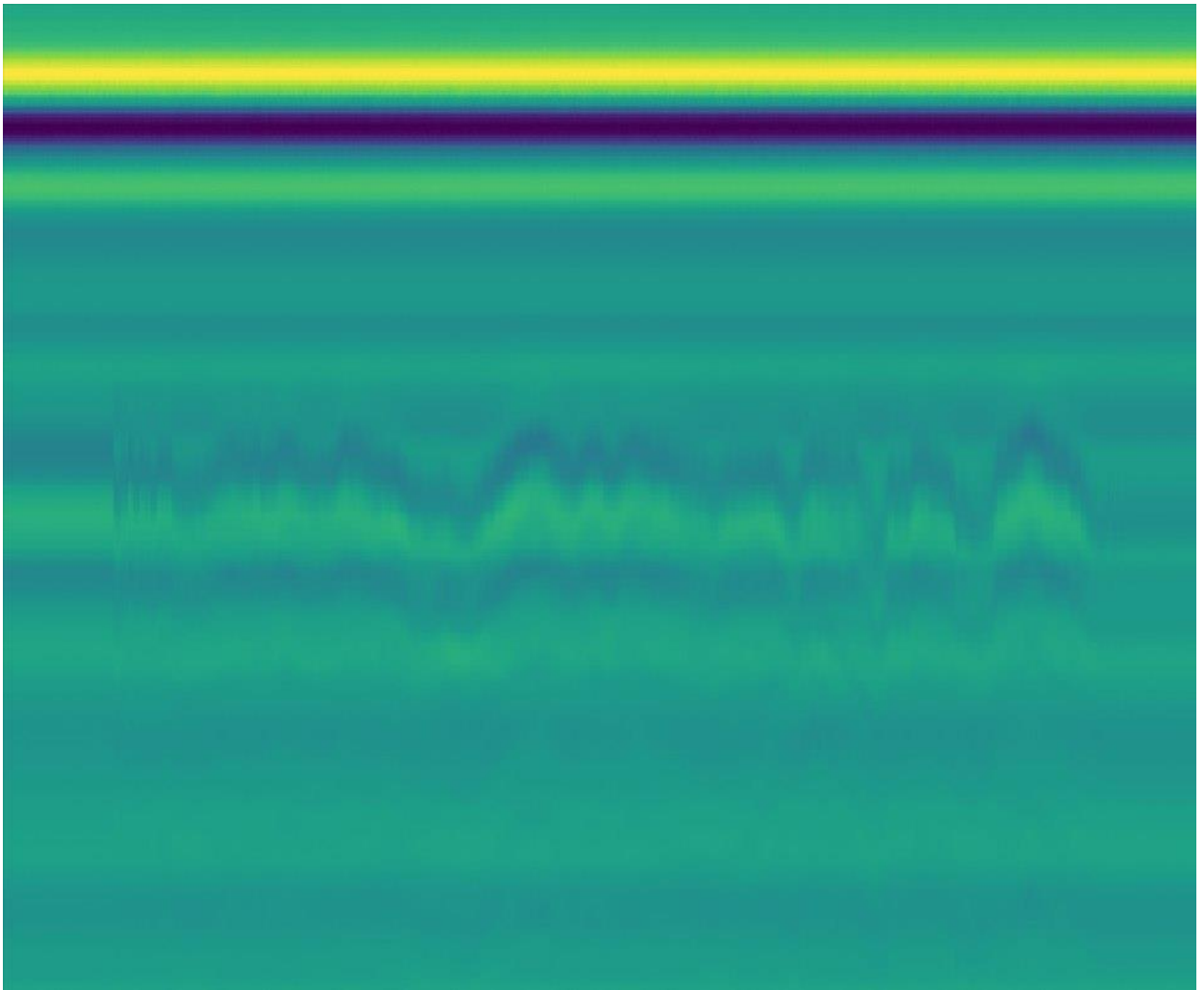


Рисунок 2.1 – Ілюстрація радарограми без попередньої обробки

По-перше треба прибрати статичне відбиття. Воно утворюється внаслідок рефлексії електромагнітного сигналу від елементів георадарної установки: антен, металевих рам, корпусу та навіть від самого передавача коли частина випромінювання повертається до приймача без заломлення в ґрунт. Оскільки ці сигнали не змінюються з часом і мають стабільну геометрію, вони формують горизонтальні лінії на радарограмі, що проходять через усе зображення. Вони мають високий рівень інтенсивності та, залишаючись незмінними незалежно від глибини чи місця зйомки, ускладнюють роботу алгоритму пошуку об'єктів.

По-друге треба прибрати відбиття від поверхні землі. Цей тип сигналу виникає, коли георадар вловлює хвилю, що відбилася від межі «повітря–ґрунт». Внаслідок різкого перепаду діелектричних властивостей на цій межі,

сигнал частково повертається назад одразу після випромінювання. В GPR-зображенні це проявляється як чітка горизонтальна смуга у верхній частині радарограми, зазвичай з високою інтенсивністю. У реальних умовах, особливо під час переміщення георадара по нерівній місцевості, висота цього сигналу на зображенні GPR не залишається постійною. Це означає, що рівень, на якому з'являється горизонтальне відбиття, різний на кожному стовпці радарограми. Внаслідок цього будь-яке глобальне або нерухоме припущення щодо положення цього шару буде неточним і потенційно призведе до втрати або спотворення важливих підповерхневих сигналів, зокрема параболічних відбиттів від об'єктів.

Тому передусім необхідно оцінити профіль поверхні, тобто визначити точку найсильнішого відбиття у кожній вертикальній трасі. Для цього аналізується кожен стовпчик зображення (тобто кожен імпульс), і в межах допустимого верхнього діапазону глибини обирається точка з найбільшою амплітудою. Вона вважається кандидатом на поверхнєве відбиття. Таким чином, отримується одномірна функція, яка описує варіацію рівня поверхні по горизонталі – профіль поверхні.

Наступним кроком є вирівнювання (flattening) зображення відповідно до цього профілю. Для кожного стовбця виконується вертикальне зміщення сигналу так, щоб відбиття від поверхні розташовувалося на одному й тому ж рівні на всьому зображенні. Це дозволяє перетворити змінну поверхню на умовно плоску. Лише після вирівнювання можна безпечно застосовувати стандартні методи фільтрації – як, наприклад, віднімання фону або високочастотне згладжування – не ризикуючи втратити корисний сигнал.

Щоб видалити ці горизонтальні фонові сигнали, застосовуються методи цифрової фільтрації. Найбільш простим і поширеним підходом є рядкова медіанна фільтрація, яка виконується по кожному стовпчику зображення окремо. Вона ґрунтується на припущенні, що горизонтальний фон має сталу або плавно змінну інтенсивність по вертикалі, тоді як параболі (відбиття від об'єктів) мають локальну форму з підвищеною контрастністю. Технічно це

реалізується шляхом обчислення медіанного значення для кожного рядка, яке потім віднімається з відповідного пікселя на зображенні. У результаті видаляється фонове значення яскравості, притаманне горизонтальному шуму.

Більш точним, але обчислювально затратним методом є застосування високочастотного гауссівського фільтра, що виділяє різкі зміни інтенсивності у вертикальному напрямку. Цей фільтр дозволяє приглушити компоненти сигналу з низькою частотою (тобто ті, що змінюються повільно по вертикалі – зокрема горизонтальний фон), зберігаючи при цьому локалізовані параболічні структури, що мають високу частотну складову. На практиці це реалізується як віднімання результату гауссівського згладжування від початкового зображення, що дозволяє підсилити деталі, приховані за рівномірним фоном [18].

2.2.2 Нормалізація та підсилення контрасту

Після усунення горизонтального фону в GPR-даних необхідно нормалізувати значення інтенсивності та підсилити контраст [19]. Ці процедури важливі для забезпечення ефективної роботи алгоритму детекції, зокрема – функції пристосованості (fitness function) в генетичному алгоритмі, яка сильно залежить від локальних змін амплітуди. Якщо GPR-зображення має надто вузький діапазон інтенсивностей або приховані контури об'єктів, то навіть при наявності правильної параболи її виявлення може бути ускладненим або неможливим [17–19].

Нормалізація методом Min-Max полягає в масштабуванні всіх пікселів зображення до заданого діапазону значень. Така лінійна трансформація дозволяє привести зображення до стандартизованого вигляду незалежно від первинного рівня сигналу, що важливо при роботі з різними наборами даних. Крім того, приведення значень до одного діапазону дозволяє уникати

упередженості алгоритму до окремих ділянок, де сигнали могли бути надмірно сильними або слабкими через технічні особливості збору даних.

Однак Min-Max нормалізація лише масштабує загальний діапазон, не впливаючи на локальні контрасти, які можуть залишатися слабкими. Для вирішення цієї проблеми застосовується метод локального контрастопідсилення – CLAHE. Він працює шляхом поділу зображення на малі підобласті, у межах яких виконується вирівнювання гистограми з обмеженням контрасту, щоб уникнути посилення шуму. Результатом такого підходу є значне покращення видимості слабконтрастних деталей, що допомагає знаходити параболічні відбиття [20].

2.2.3 Фільтрація шуму

Дані, отримані за допомогою георадару, містять шумові артефакти, які можуть бути викликані як зовнішніми джерелами, так і внутрішніми особливостями самого приладу. Ці шуми проявляються у вигляді випадкових варіацій інтенсивності пікселів і можуть маскувати та спотворювати параболічні сигнатури, що значно знижує точність алгоритму автоматичного виявлення об'єктів [6, 21]. Для фільтрації таких артефактів використовуються методи гауссівського згладжування та високочастотної фільтрації. Гауссівське згладжування полягає в згладженні зображення за допомогою згортки з гауссівським ядром. Це дозволяє приглушити випадкові коливання інтенсивності, зменшити візуальний шум та підкреслити більш сталу структуру сигналів. Гауссівський фільтр дає змогу «пом'якшити» різкі переходи, не знищуючи при цьому повністю ключові контури, особливо якщо правильно підібраний радіус ядра. Значення σ (середньоквадратичного відхилення) підбирається таким чином, щоб зберегти форму параболічних сигнатур, які, хоча й не мають ідеально різких країв, але потребують певної чіткості [22–23].

Наступний метод, високочастотна фільтрація, застосовується для виділення лише тих компонент зображення, які змінюються швидко, тобто мають локальні контрастні переходи. Для даної проблеми підхід реалізується шляхом побудови згладженого низькочастотного зображення за допомогою того ж гауссівського фільтра, а потім віднімання цієї розмитої версії від оригіналу. Таким чином, на виході залишається лише високочастотна складова сигналу, що відповідає точкам із різкими змінами інтенсивності – саме вони найбільш характерні для країв парабол. Така обробка допомагає ще більше відокремити релевантні для детекції області від загального шуму та зменшує навантаження на генетичний алгоритм, який і надалі працює лише з інформаційно насиченими зонами зображення.

Застосування згладжування та високочастотного виділення дозволяє досягти компромісу між зменшенням випадкових завад і збереженням деталей. Це необхідно в умовах, коли об'єкти можуть бути розташовані на складному або зашумленому фоні, як це часто трапляється в реальних георадарних вимірюваннях. Завдяки цим методам фільтрації GPR-зображення готується до подальшого етапу – виявлення параболічних структур за допомогою еволюційного алгоритму, який після фільтрації працює з набагато точнішими даними.

2.2.4 Кадрування області інтересу

У георадарних зображеннях особливістю є те, що верхня частина скану часто містить статичне відбиття, яке проявляється як декілька горизонтальних смуг з великою амплітудою. Це – технічний артефакт, що виникає в результаті відбиття сигналу від частин самого пристрою, і не несе інформації про підземні об'єкти. Включення цієї області в аналіз лише ускладнює роботу алгоритму, адже у такому випадку необхідно обробити більшу частину скану.

До того ж навіть після фільтрації там можуть залишитись незначні артефакти. У таких випадках застосовують кадрування області зацікавленості (ROI).

Кадрування це своєрідний фільтр уваги алгоритму, який дозволяє сфокусуватись на тих частинах скану, де найімовірніше знаходяться параболи, та ігнорувати інші зони, які є зашумленими, зайвими або технічно непридатними для аналізу. Це підвищує як швидкодію системи, так і точність виявлення об'єктів через меншу площу для аналізу.

Практично це виконується шляхом обтинання фіксованої кількості верхніх рядків скану – 10–15% висоти зображення, залежно від глибини проникнення сигналу та конфігурації пристрою.

Наступним кроком є виділення глибших областей. Знаючи параметри георадарної установки, можна припустити, на якій мінімальній відстані від антен-приймачів за фізикою процесу найімовірніше розташовані параболічні відбиття від підземних об'єктів. У цій зоні відображення зазвичай слабші, проте саме вони мають параболічну форму, що й становить головний інтерес для алгоритму. У випадку реальних даних, як було показано у попередньому розділі, параболи зазвичай розміщуються у нижній половині зображення, отже кадрування з фокусом на цю зону дозволяє прибрати зайвий візуальний шум у вигляді деяких горизонтальних артефактів, підземних перешкод або обривів на глибині [3].

При надходженні даних з георадару у реальному часі є сенс використати горизонтальне кадрування, коли зображення розділяється на декілька сегментів по ширині (наприклад, вікнами шириною 64 або 128 пікселів) [10]. Це також актуально в задачах послідовного або багатопоточкового аналізу, де генетичний алгоритм застосовується окремо до кожної ділянки, дозволяючи виявити кілька об'єктів незалежно одне від одного. За один крок вікно має бути зміщене на половину довжини сегменту, щоб врахувати відбиття на границях сегменту. Це зменшує розмірність задачі оптимізації, і дозволяє адаптивно підлаштовувати параметри обробки до локальних характеристик сигналу [10, 24].

2.3 Налаштування алгоритму згідно з умовами задачі

2.3.1 Пояснення вибору генетичного алгоритму

Існує декілька видів еволюційних алгоритмів: класичний генетичний алгоритм (GA), еволюційна стратегія (ES) та диференціальна еволюція (DE). Попри те, що всі три підходи належать до одного класу стохастичних оптимізаторів і здатні виконувати пошук у складному, неперервному або дискретному просторі, GA є найбільш перспективним для вирішення даної задачі.

Генетичний алгоритм відзначається своєю простотою у реалізації. Його основні компоненти – популяція, селекція, кросовер та мутація – є концептуально зрозумілими та легко кодуються мовами програмування високого рівня. Навіть базовий варіант алгоритму без складних стратегій адаптації параметрів здатен знаходити параболу у шумовому середовищі з достатньо високою точністю.

У порівнянні з цим, еволюційна стратегія (ES), хоча й подібна за принципом, є більш математично строгою і передбачає використання стратегічних параметрів, таких як ковариаційна матриця (у варіантах SMA-ES), адаптивні множники кроків та інші. Її основна перевага – здатність до швидкої локальної збіжності у високовимірних просторах, однак у випадку даної задачі, простір параметрів складається лише з трьох змінних: координати вершини параболи та її коефіцієнт кривизни (x_0, y_0, a) . У такому низьковимірному випадку застосування складної стратегії оптимізації виглядає недоцільним: виграш у точності буде мінімальним, зате реалізація значно ускладниться.

Диференціальна еволюція (DE), також вважається одним з ефективних еволюційних методів, особливо при мінімізації складних функцій. Її ключова особливість – використання різниці між двома випадковими індивідами для генерації нових рішень. DE часто показує кращу збіжність, ніж класичний GA, на стандартних функціях оптимізації, однак ця перевага проявляється

насамперед у задачах, де є гладка й диференційовна функція втрат. У випадку з аналізом GPR-зображень, функція якості (fitness) є неформальною, нерегулярною, з великою кількістю шуму та локальних максимумів. Тут стохастичний характер GA має перевагу, оскільки кросовер і мутація генерують більш різноманітні рішення, не покладаючись на градієнт.

З погляду швидкодії, усі три методи мають приблизно однакову асимптотичну складність – кожне покоління вимагає обчислення значення функції для кожного індивіда.

2.3.2 Основні поняття генетичного алгоритму

Генетичний алгоритм (GA) як метод оптимізації базується на біологічній еволюції, зокрема на процесах природного добору. Його ефективність значною мірою залежить від правильної постановки задачі в термінах еволюційної теорії. Ключовими поняттями даного алгоритму є популяція, покоління та хромосома.

Хромосома – це основна структурна одиниця у генетичному алгоритмі, яка містить набір параметрів, що описують одне можливе рішення задачі. В даному випадку набір параметрів це (x_0, y_0, a) , він повністю визначає форму параболічної кривої. У генетичному алгоритмі кожна така хромосома є кандидатом на відповідь і аналізується за допомогою функції пристосованості.

Популяція – це набір хромосом (тобто множина можливих рішень), які одночасно аналізуються на кожному кроці алгоритму. Популяція задається випадковим чином у першому поколінні, а потім поступово оновлюється через процеси селекції, кросоверу та мутації. Велика популяція дозволяє алгоритму охопити ширший простір пошуку та уникати передчасної збіжності до локального максимуму, однак збільшує обчислювальні витрати. З іншого боку, занадто мала популяція зменшує різноманітність рішень і ризикує залишитися в неефективній області простору параметрів.

Покоління – це одна ітерація алгоритму, яка складається з оцінювання всіх хромосом популяції, вибору найбільш пристосованих, та генерації нових хромосом на основі старих. Після кожного покоління старі хромосоми або зникають, або трансформуються, а нові набувають шансів зайняти їх місце. Кількість поколінь визначає, скільки разів алгоритм повторює процес еволюції. На кожному кроці відслідковується найкраще значення функції пристосованості та зберігаються найуспішніші хромосоми.

2.3.3 Функція пристосованості

Функція пристосованості (fitness function) визначає, наскільки добре окрема особина алгоритму відповідає ознакам підземного об'єкта в радіолокаційному зображенні. Базовий підхід полягає у виборі функції, що обчислює суму квадратів інтенсивностей пікселів вздовж дуги параболи-кандидата. Такий підхід дає перевагу тим параболом, що проходять через контури з високою контрастністю – тобто типові для відбиттів сигналу від металевих об'єктів або кабелів.

Сума квадратів замість простої суми дозволяє зробити внесок сильних сигналів (тобто яскравих точок) суттєво більшим у загальному значенні пристосованості. Таким чином, параболи, що проходять через сукупності яскравих пікселів (сигнали від об'єктів), отримують значно більшу оцінку, ніж ті, що накладаються лише на шум або тіньові області.

Однак якщо дві параболи проходять через зони однакової інтенсивності, але одна з них охоплює більше пікселів з високою інтенсивністю то необмежена сума квадратів інтенсивностей призведе до завищеної оцінки для даної параболи – навіть якщо вона не повністю відповідає реальній параболі на радарограмі.

Щоб вирішити цю проблему, впроваджується нормалізація після обчислення суми квадратів інтенсивностей [25]. Ця сума ділиться на кількість

пікселів N , які фактично були враховані під час обчислення (тобто тих, що лежали всередині меж зображення). Таким чином, функція пристосованості фактично відображає середню інтенсивність уздовж параболи, а не просто її загальну «яскравість». Це дозволяє порівнювати між собою параболи різної довжини, незалежно від того, скільки пікселів охоплює кожна з них.

Така нормалізація також дозволяє адаптувати алгоритм до зображень із неоднорідною освітленістю або контрастом. Наприклад, на одному знімку об'єкти можуть бути яскравішими, а на іншому – менш контрастними. За відсутності нормалізації алгоритм схилився б до вибору найяскравішої області, навіть якщо вона не є параболою [26].

Крім того, при багатократному пошуку кількох парабол у одному зображенні нормалізація дозволяє справедливо порівнювати *fitness* різних кандидатів і вводити єдиний поріг для зупинки пошуку.

2.3.4 Штраф за малу кривизну параболи

Інколи виникає ситуація, коли майже горизонтальні параболи з коефіцієнтом кривизни близьким до нуля можуть частково збігатися зі слабким шумом чи випадковими змінами інтенсивності в зображенні. У таких випадках алгоритм може помилково надати високу оцінку даним параболом. Щоб уникнути цієї проблеми, до функції пристосованості вводиться штраф за малу кривизну. Це досягається за допомогою множника-штрафу, який залежить від абсолютного значення коефіцієнта кривизни. Чим менше є значення цього коефіцієнта, тим більш пласкою є парабола і тим більший накладається штраф.

Такий підхід дозволяє штучно знижувати можливість вибору кандидатів із надто малою кривизною, бо ймовірніше такі кандидати є випадковими шумовими структурами або залишками горизонтальних артефактів (наприклад, відбиття від поверхні ґрунту).

2.3.5 Відбір

Оператор відбору визначає, які особини (параметри потенційних парабол) із поточної популяції будуть допущені до створення наступного покоління. В якості такого оператора використовується стохастичний метод відбору з імовірнісним зважуванням, відомий як «roulette wheel selection» або пропорційний відбір. Його суть полягає в тому, що кожна особина має ймовірність бути обраною пропорційно до свого значення функції пристосованості.

Таким чином, особини, які краще описують параболічну структуру на зображенні, мають більші шанси залишити наступне покоління, але навіть менш придатні рішення мають ненульову ймовірність бути вибраними – це запобігає передчасній конвергенції до локального оптимуму.

У разі, коли всі значення функції пристосованості близькі до нуля (наприклад, при дуже низькому контрасті зображення), застосовується рівномірний відбір: всі особини мають однакову ймовірність бути обраними, що гарантує продовження роботи алгоритму.

На кожному поколінні з популяції випадковим чином обирається k особин із повторенням (тобто одна особина може бути обрана кілька разів), відповідно до вищезгаданих ймовірностей. Це дозволяє зберігати розмір популяції стабільним між поколіннями. Такий механізм ефективно поєднує експлуатацію – використання вже знайдених рішень – та дослідження нових варіантів простору рішень.

Завдяки стохастичній природі цього підходу, навіть неідеальні рішення не відкидаються повністю. Це сприяє збереженню різноманіття в популяції, а отже – підвищує ймовірність знайти правильні параметри парабол навіть у випадку слабких або частково зашумлених об'єктів на сканах. Також це дозволяє на зупинитися на локальних екстремумах, а повністю аналізувати радарограму.

2.3.6 Схрещування

Для операції схрещування (кросоверу) було обрано метод арифметичного схрещування, який базується на лінійній інтерполяції між двома батьківськими особинами. Такий підхід є простим, ефективним і добре підходить для задач, де особини представлені у вигляді векторів числових параметрів [27]. Кожна нова особина-нащадок створюється як зважене середнє між двома батьками з використанням випадково згенерованого коефіцієнта.

2.3.7 Мутація

Оператор мутації забезпечує генетичне різноманіття популяції та дозволяє уникнути передчасної збіжності алгоритму до локального екстремуму. Мутація застосовується до кожної нової особини з певною фіксованою ймовірністю. При настанні події мутації кожен із параметрів особини змінюється за допомогою додавання гауссівського шуму з відповідним стандартним відхиленням. Це дозволяє варіативно коригувати параметри особин у довільному напрямку, залишаючи при цьому розв'язки всередині допустимого простору.

Водночас імовірність мутації встановлюється не надто високою, щоб не зруйнувати уже знайдені коректні рішення, але й не надто низькою – щоби уникнути застою в популяції. Варто зауважити, що мутація особливо ефективна на пізніх етапах еволюції, коли більшість особин стали подібними, і лише випадкові зміни можуть вивести процес оптимізації з локального максимуму та допомогти знайти точніше розв'язання.

Стохастичний характер оператора забезпечує дослідження нових областей простору параметрів, які могли бути пропущені операторами селекції та схрещування.

2.3.8 Елітизм

Елітизм це механізм генетичного алгоритму, який гарантує збереження найкращих знайдених рішень при переході до наступного покоління. Він дозволяє не втратити найбільш успішні параметри парабол, які демонструють високу відповідність сигналам на зображенні, навіть якщо під час мутації або кросоверу нові особини виявляються менш ефективними.

Механізм елітизму виглядає так: перед формуванням нового покоління проводиться сортування всієї популяції за значенням функції пристосованості (fitness), після чого кілька (одна або кілька) найкращих особин копіюються без змін у нову популяцію. Таким чином, ці особини автоматично зберігаються для подальшого використання. Це гарантує, що якість розв'язку не погіршиться в наступному поколінні, навіть за умов значного стохастичного шуму або агресивної мутації.

Цей підхід забезпечує сталість результату в умовах випадковості, а також прискорює збіжність алгоритму, оскільки відомо, що принаймні один з високоякісних розв'язків завжди присутній [28]. Без елітизму існує ризик, що алгоритм загубить вже знайдені успішні параметри, якщо випадково створені менш вдалі особини отримають перевагу в поточному поколінні.

2.3.9 Параметри алгоритму

Розмір популяції – це кількість параметричних комбінацій парабол, які одночасно розглядаються на кожному кроці генетичного алгоритму. Цей параметр визначає об'єм простору рішень, що досліджується за одну ітерацію. Якщо популяція занадто мала, існує ризик ранньої конвергенції до локального максимуму – тобто алгоритм знайде «слабке» рішення й припинить пошук кращого. Якщо ж розмір занадто великий, це призводить до надмірного навантаження на ресурси і подовження часу обчислень.

Кількість поколінь визначає, скільки разів відбудеться процес селекції, схрещування та мутації особин у популяції. Це кількість «ітерацій навчання» алгоритму. При кожному поколінні особини з вищою функцією пристосованості мають вищі шанси залишити потомство – таким чином, з покоління в покоління еволюційний тиск спрямовує популяцію до кращого рішення. Мале значення може бути недостатнім для стабільної локалізації параболи, особливо в зашумлених даних. Велика кількість може бути надмірною, особливо якщо максимум fitness досягається раніше.

Ймовірність мутації регулює частоту випадкових змін параметрів у нащадках. Цей механізм дозволяє уникнути передчасного сходження до локального максимуму та забезпечує достатнє різноманіття рішень. Без мутацій популяція швидко вироджується до схожих рішень, які можуть бути далекі від глобального максимуму.

Надмірно висока ймовірність веде до хаотичних коливань і руйнування вже знайдених гарних рішень. Надто мала підвищує ризик конвергенції до локального максимуму.

2.3.10 Адаптація алгоритму для пошуку декількох об'єктів на радарограмі

В реальних даних георадарного зондування часто присутні не одна, а кілька параболічних сигнатур, що відповідають різним підземним об'єктам. Для виявлення всіх таких об'єктів необхідно застосовувати ітеративний підхід до обробки зображення, в якому після кожного акту детекції зображення модифікується, щоб уникнути повторного знаходження тієї ж самої параболи або артефактів, що її оточують [28–29].

На першому кроці ітерації застосовується генетичний алгоритм до поточного зображення. Алгоритм виконує пошук параметрів однієї параболи, які дають найвищу оцінку функції пристосованості. Після завершення певної

кількості поколінь та мутацій, алгоритм повертає найкращу особину, яка вважається знайденою параболою. Якщо функція пристосованості цієї особини перевищує встановлений поріг, ця парабола вважається валідною і підлягає подальшій обробці.

Після успішного знаходження однієї параболи, наступним кроком є маскування зони, де ця парабола розташована. Для цього визначається область впливу навколо вершини x_0 : вибирається певна кількість колонок ліворуч та праворуч від цієї координати. Всі значення інтенсивностей у цих колонках зануляються, тим самим усуваючи зображення знайденої структури з подальшого аналізу. Такий підхід запобігає повторному знаходженню тієї ж самої параболи або її частин у наступних ітераціях, оскільки область, де функція пристосованості була високою, більше не містить корисної інформації.

Після маскування зображення, алгоритм повторно запускає генетичний пошук вже на оновленій версії зображення. Оскільки одна з парабол була видалена, інші сигнатури тепер мають шанс отримати вищі значення функції пристосованості. Цей процес знову завершується детекцією ще однієї параболи, яка також маскується. Таким чином, цикл «детекція – маскування – повторний пошук» повторюється доти, доки існують потенційні об'єкти, чий значення функції пристосованості більше за певний поріг [30].

2.3.11 Фінальний вигляд алгоритму

Блок-схема що ілюструє повний процес пошуку об'єктів на радарограмі показана на рисунку 2.2. В ній враховано попередню обробку скану, а також ітеративний пошук об'єктів з маскуванням областей зображення, на яких ці об'єкти були виявлені. Вхідною точкою є отримання скану (радарограми), умовою завершення є занадто мале значення функції пристосованості.

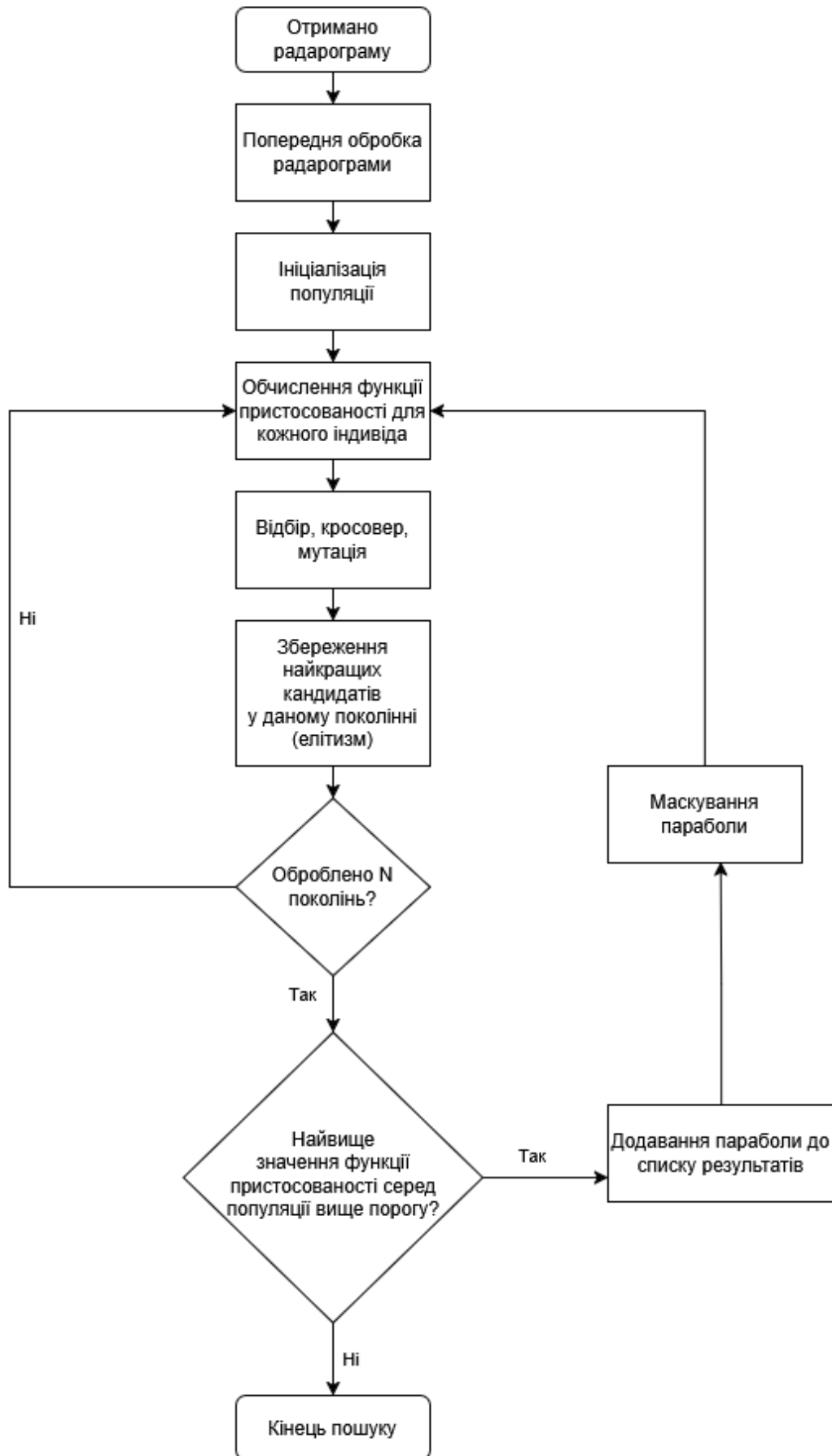


Рисунок 2.2 – Блок-схема алгоритму

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

3.1 Вибір середовища програмної реалізації

В рамках розробки вебзастосунку для пошуку об'єктів на невеликій глибині було використано набір інструментів для реалізації різних компонентів системи. Архітектура застосунку побудована за принципом поділу на клієнтську та серверну частини.

Клієнтська частина реалізована мовою JavaScript. Для побудови інтерфейсу користувача застосовано бібліотеку React, що забезпечує компонентну архітектуру розробки. Такий підхід дозволив представити кожен елемент інтерфейсу (список сканів, форму налаштувань алгоритму, панель візуалізації) у вигляді окремого реактивного компонента з автоматичним оновленням при зміні стану.

Збірка та запуск фронтенду здійснюється за допомогою білдера Vite, який використовує модульну структуру ECMAScript та забезпечує швидкий старт проєкту і миттєве оновлення модулів без повної перебудови системи.

Для забезпечення навігації між сторінками використано бібліотеку react-router-dom, що надає засоби маршрутизації у React-застосунках. Це дозволило реалізувати перехід між розділами «Список сканів» та «Параметри алгоритму» без перезавантаження сторінки.

Обмін даними із серверною частиною реалізовано через бібліотеку axios – HTTP-клієнт для JavaScript. За допомогою даної бібліотеки виконуються GET та POST запити до API-бекенду, обробляються відповіді та помилки, налаштовуються заголовки запитів.

Керування зовнішніми залежностями здійснюється через npm – менеджер пакетів Node.js. Цей інструмент використовується для додавання та оновлення бібліотек, а також налаштування скриптів запуску, тестування та білдування застосунку.

Для оформлення інтерфейсу застосовано технологію CSS Modules, що забезпечує ізоляцію стилів між компонентами, уникнення конфліктів стилів та підтримку масштабованості застосунку. На відміну від глобального CSS, модульний підхід дозволяє імпортувати стилі як об'єкти в JavaScript та використовувати їх у компонентах.

Для реалізації двосторонньої комунікації в реальному часі використано бібліотеку `socket.io-client`.

Клієнтська частина, розроблена з використанням бібліотеки React, відповідає за взаємодію з користувачем, включаючи відображення списку сканів, налаштування параметрів еволюційного алгоритму та візуалізацію результатів детекції. Взаємодія із сервером відбувається через REST-API (Axios) для отримання метаданих сканів і виклику процесу обробки, а також через WebSocket (`socket.io-client`) для отримання логів виконання алгоритму пошуку об'єктів у режимі реального часу.

Серверна частина вебзастосунку реалізована мовою Python. Для створення бекенду обрано Flask – мікрофреймворк, що забезпечує розробку RESTful API з необхідною гнучкістю побудови проєкту та можливістю масштабування.

З метою ізоляції залежностей та забезпечення відтворюваності середовища розробки використано програмний модуль `venv`, що входить до комплекту інтерпретатора Python. За допомогою цього модуля створено віртуальне середовище для локального встановлення бібліотек проєкту без модифікації системних пакетів. Такий підхід забезпечує сумісність версій та запобігає виникненню конфліктів між залежностями під час розробки та розгортання застосунку.

Зовнішні бібліотеки бекенду встановлені через стандартний Python менеджер пакетів `pip`. Після встановлення їхні версії збережено в файлі `requirements.txt`, що дозволяє відтворити середовище на іншому комп'ютері або сервері.

Для забезпечення доступу до серверних ресурсів з клієнтського застосунку, що працюють на різних портах, використано бібліотеку flask-cors з налаштуванням політики CORS.

Такі налаштування забезпечують відкритий доступ до маршрутів для всіх доменів, що є прийнятним варіантом у рамках однокористувацького застосунку з можливістю подальшого коригування політики доступу.

Модуль попередньої обробки зображення та пошуку об'єктів з використанням генетичного алгоритму також реалізовано мовою Python. На етапі попередньої обробки георадарних сканів застосовано бібліотеки NumPy, OpenCV та SciPy.

Бібліотека NumPy використана для роботи з двовимірними масивами, що представляють радарні зображення. За її допомогою виконуються операції масштабування, обтинання, обчислення статистик та математичні операції, зокрема нормалізація значень пікселів.

Для роботи з зображеннями застосовано бібліотеку OpenCV. Вона дозволяє зчитувати та зберігати скан, змінювати контрастність, видаляти фон та обробляти шуми. Одним із методів поліпшення зображення є видалення смуги, що відповідає поверхні (горизонтальна область у верхній частині скану), а також масштабування зображення до діапазону значень пікселів для забезпечення сумісності з алгоритмами.

Бібліотека SciPy використана у випадках, де потрібна згладженість або частотна фільтрація, зокрема її модуль scipy.ndimage. Зокрема, застосування гауссівського фільтру дозволяє зменшити шум для більш якісної роботи генетичного алгоритму.

Середовищем розробки для фронтенду та бекенду обрано редактор Visual Studio Code з розширеннями для підтримки синтаксису JavaScript, Python, форматування коду, автодоповнення та навігації по проєкту. Це є безкоштовною альтернативою популярним IDE, таким як WebStorm, PyCharm. Також, на відміну від інших рішень, Visual Studio Code дозволяє працювати з

різними частинами вебзастосунку. Pycharm, у свою чергу, працює тільки з мовою Python, а WebStorm – тільки з клієнтською частиною.

Для контролю версій використано систему Git, що забезпечило збереження змін у кодовій базі та можливість створення окремих гілок для розробки нових функцій. Регулярні коміти дозволили відстежувати історію розробки та повертатись до попередніх версій за необхідності. Для виключення зайвих службових файлів було використано файл `.gitignore`. Прикладами таких файлів є кеш CPython та завантажені через Node Package Manager модулі.

Для тестування API-сервісів бекенд-частини застосовано Postman – популярний інструмент для тестування API, який дозволяє розробникам надсилати HTTP-запити до серверів, переглядати відповіді, автоматизувати тестування та документувати програмні інтерфейси бекенду. Postman значно спрощує взаємодію з веб-сервісами на етапі їх розробки. Користувач може задавати методи запитів, додавати заголовки, параметри і тіло кожного запиту, а також переглядати відповіді у зручному форматі. Крім того, Postman підтримує створення колекцій запитів, написання автоматичних тестів за допомогою JavaScript, а також інтеграцію з CI/CD-процесами.

За його допомогою протестовано логіку обробки маршрутів, а також коректність обробки параметрів і повернення відповідей у форматі JSON.

3.2 Програмна реалізація

3.2.1 Створення бекенду

Головним файлом бекенд частини є `app.py`, там відбувається ініціалізація Flask серверу та додавання власних шляхів маршрутизації (`routes`). Основна логіка бекенду розташована в директорії `scans` (рис. 3.1).

Файл `.python-version` містить версію інтерпретатора, а `requirements.txt` містить залежності, які встановлюються через пакетний менеджер `pip`.

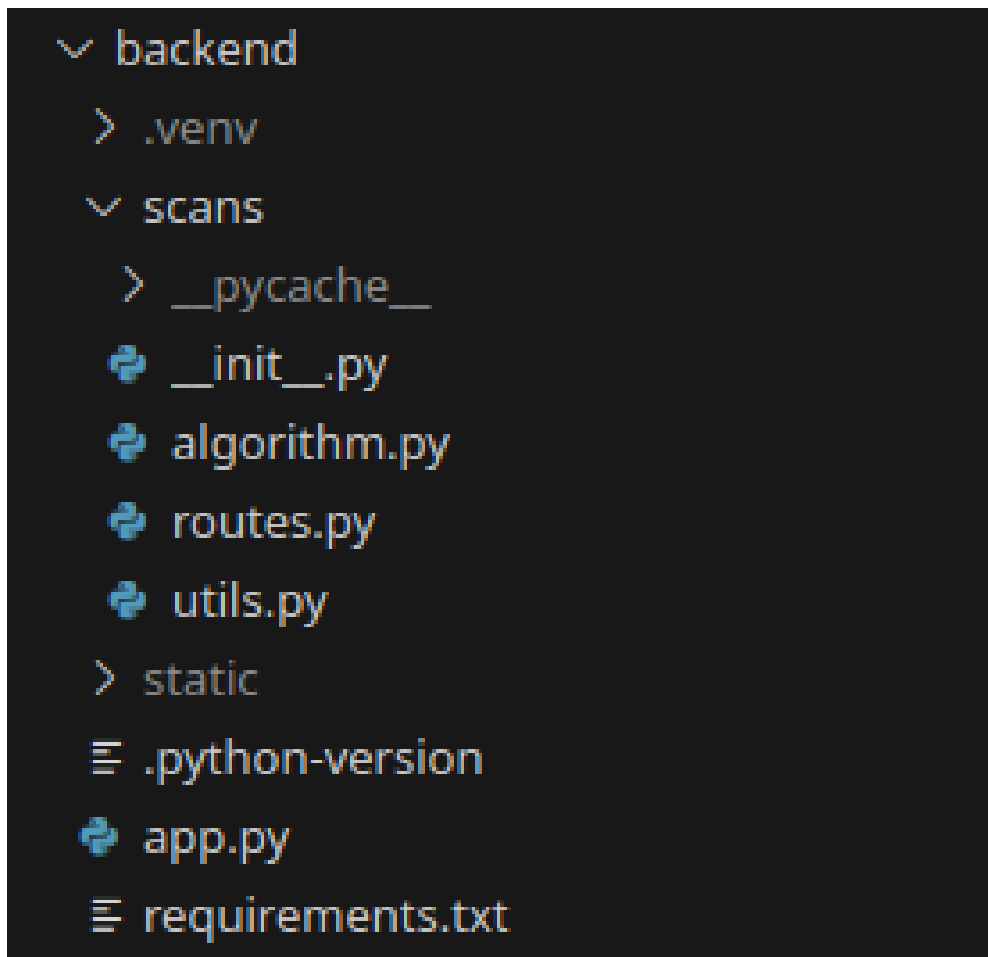


Рисунок 3.1 – Структура файлів бекенду

Для зберігання даних було обрано підхід на основі файлової системи. Це обумовлено специфікою збереження даних у даному застосунку, відсутністю необхідності підтримки декількох користувачів та використанням застосунку у локальній мережі разом з іншим програмним забезпеченням для роботи з радарною системою. Більша частина програмного забезпечення для аналізу даних георадару експортує та імпортує скани у вигляді файлів, тож раціональним є використання саме такого формату. Також файлова організація даних спрощує процес розробки, тестування та впровадження системи, оскільки не потребує конфігурації та підтримки складної інфраструктури баз даних. Завдяки модульності застосунку, в майбутньому в нього може бути інтегрована база даних, якщо необхідно. Вигляд директорії, де зберігаються дані застосунку подано на рисунку 3.2.

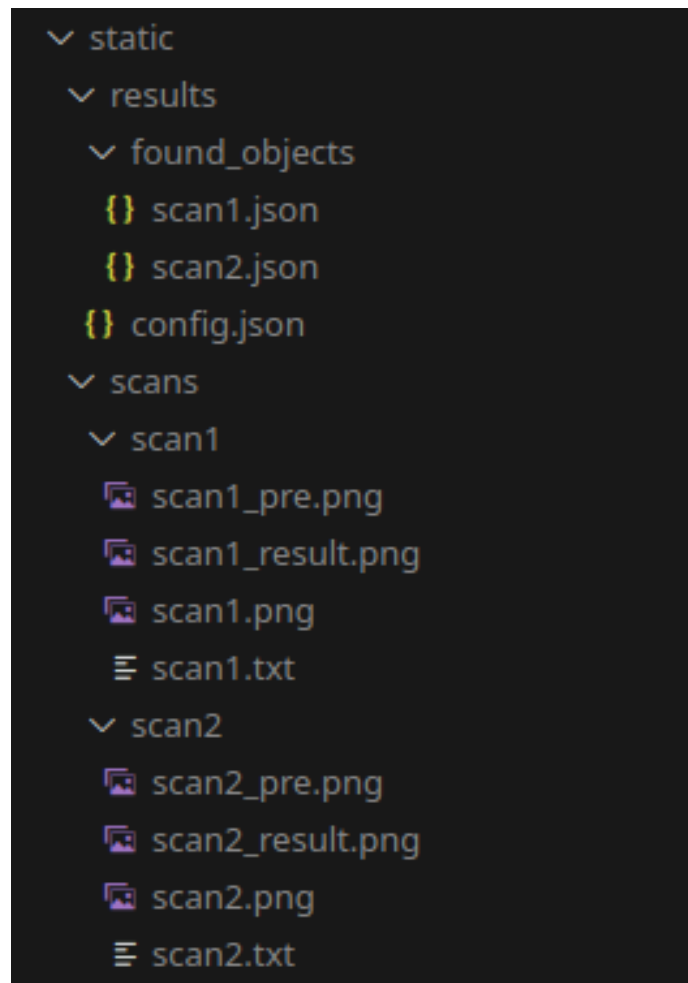


Рисунок 3.2 – Структура сховища даних у файловій системі

В директорії `static/scans` розташовані скани з радару в окремих папках. `Scan1/scan1.txt` представляє собою двовимірний масив, записаний текстовим форматом. Це є поширений спосіб збереження даних серед інших програм для роботи з науковими обчисленнями, навіть в бібліотеці NumPy імплементовано функції для завантаження та збереження масивів в такому форматі.

Файли формату `png` представляють собою зображення, згенеровані на основі георадарного скану. Це необхідно для ілюстрації та перевірки роботи алгоритму оператором. `Scan1.png` це `heatmap` зображення скану без попередньої обробки. `Scan1_pre.png` це зображення після попередньої обробки, тут оператор може самостійно проаналізувати його, наприклад, для того, щоб підлаштувати алгоритм до особливостей досліджуваного середовища. `Scan1_result.png` містить зображення з нанесеними контурами

парабол, що наочно демонструють знайдені об'єкти. Тут оператор може впевнитися в коректності роботи алгоритму.

Конфігурація алгоритму записана в файлі `config.json`. В директорії `found_objects/` збережені файли у форматі JSON, разом з координатами знайдених об'єктів відносно самого скану та метаданими.

Застосунок обробляє певну кількість REST методів. За допомогою них фронтенд взаємодіє з бекендом. Імплементовано методи отримання списку сканів (`/api/scans/`), деталей скану (`/api/scans/<id>`), методи для отримання та оновлення конфігурації (`/api/config`) та метод для запуску алгоритму (`/api/process`).

Для тестування розробленого API було використано програмне забезпечення Postman. На рисунку 3.3 показано запит для отримання списку доступних для обробки сканів.

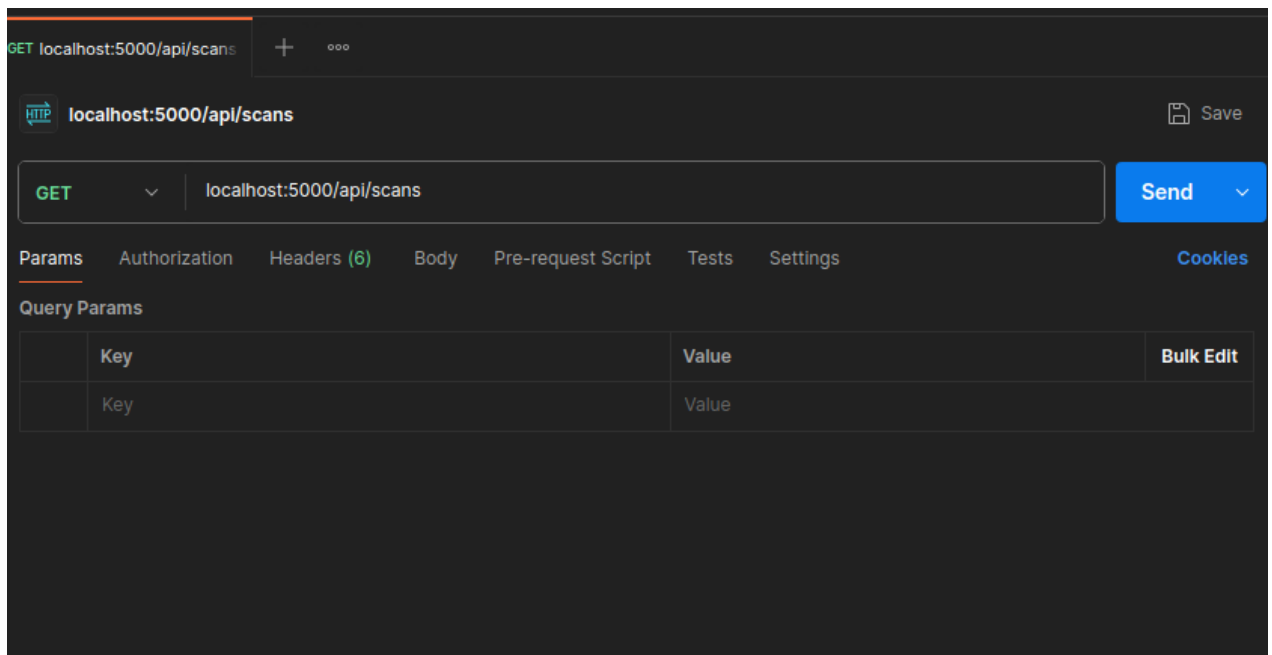


Рисунок 3.3 – Запит на отримання списку сканів

Результат запиту показано на рисунку 3.4. Серед результатів можна побачити назву скану («scan1», «scan2»), його ідентифікатор, та дату завантаження до системи. Ідентифікатор є унікальним для кожного скану, тому при однакових іменах сканів, наприклад, коли для збереження сканів

використовуються різні директорії, ідентифікатор буде відрізнятися від імені скану.

```
{
  "filename": "scan2",
  "id": "scan2",
  "uploaded_at": "2025-05-18T09:05:57.918567"
},
{
  "filename": "scan1",
  "id": "scan1",
  "uploaded_at": "2025-05-17T23:55:35.554282"
}
```

Рисунок 3.4 – Результат виконання запиту

На рисунку 3.5 показано виконання запиту для отримання деталей скану «scan1».

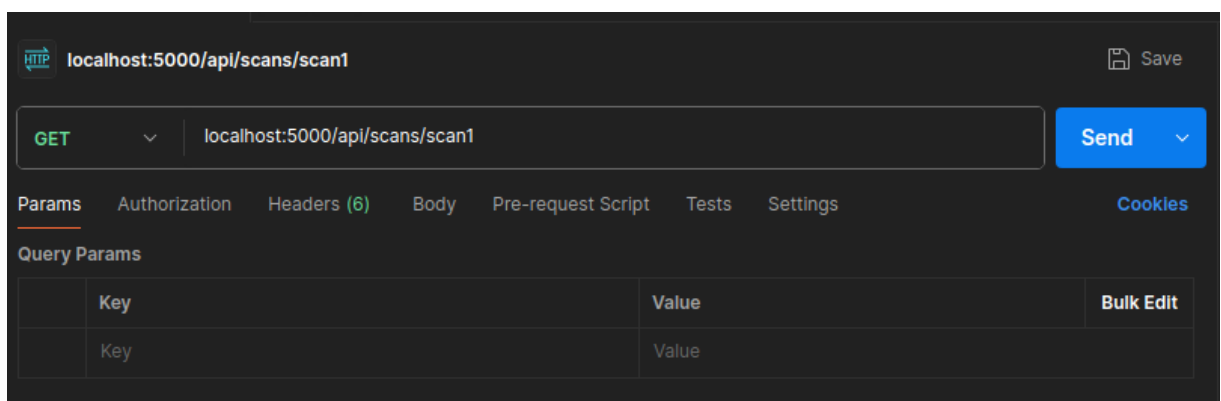


Рисунок 3.5 – Запит на отримання детальної інформації про скан

Деталі сканів включають в себе координати знайдених об'єктів відносно самого скану, а також посилання на зображення кожного скану (рис 3.6).

```
"filename": "scan1",
"id": "scan1",
"parabolas": [
  {
    "a": 0.0416151928531517,
    "fitness": 8398.606557377048,
    "x0": 167.7362213840657,
    "y0": 68.57810351419268
  },
  {
    "a": 0.04380050788177632,
    "fitness": 6946.639344262295,
    "x0": 342.1702393308343,
    "y0": 100.27288165860021
  },
  {
    "a": 0.005894626692215762,
    "fitness": 2873.0163934426228,
    "x0": 264.9990793263073,
    "y0": 106.00222396842422
  }
],
"preprocessed_url": "http://localhost:5000/static/scans/scan1/scan1_pre.png",
"raw_url": "http://localhost:5000/static/scans/scan1/scan1.png",
"result_url": "http://localhost:5000/static/scans/scan1/scan1_result.png"
```

Рисунок 3.6 – Результат виконання запиту

На рисунку 3.7 показано запит для отримання конфігурації алгоритму.

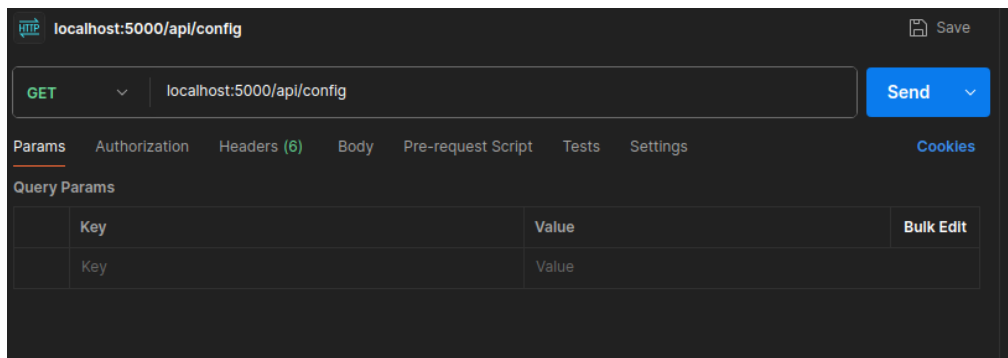


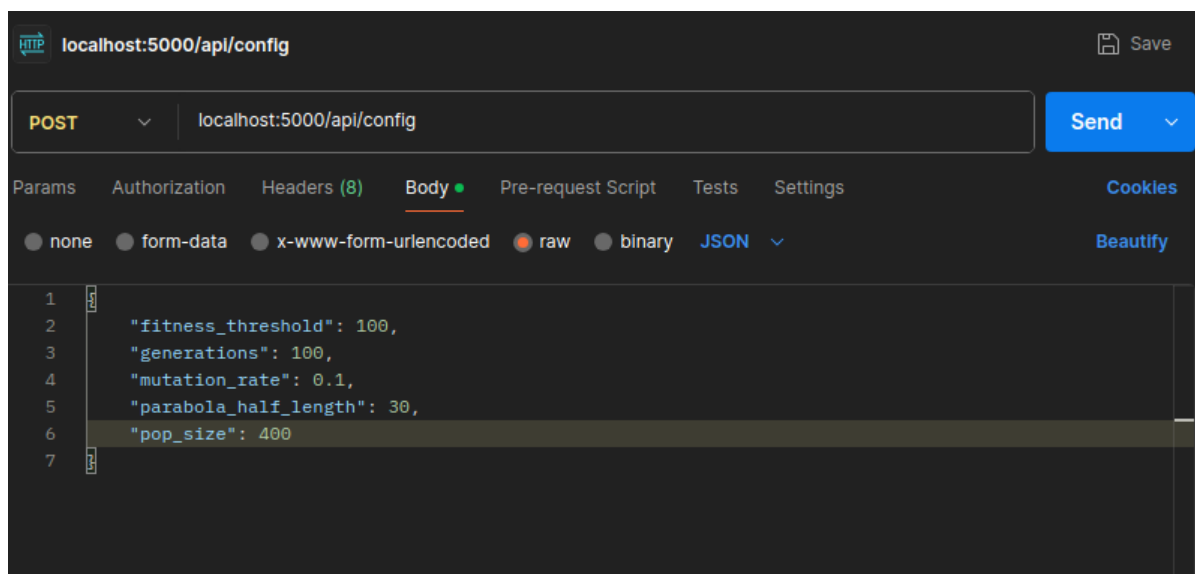
Рисунок 3.7 – Запит на отримання конфігурації

На рисунку 3.8 показано результат запиту. Параметри, які відображені на даному рисунку, можуть бути розширені у майбутньому, в залежності від вимог користувачів.

```
"fitness_threshold": 100,  
"generations": 100,  
"mutation_rate": 0.1,  
"parabola_half_length": 30,  
"pop_size": 500
```

Рисунок 3.8 – Результат виконання запиту

Далі на рисунку 3.9 показано запит на зміну збереженої конфігурації алгоритму.



The screenshot shows a REST client interface with the following details:

- URL: localhost:5000/api/config
- Method: POST
- Body (JSON):

```
1  
2 "fitness_threshold": 100,  
3 "generations": 100,  
4 "mutation_rate": 0.1,  
5 "parabola_half_length": 30,  
6 "pop_size": 400  
7
```

Рисунок 3.9 – Запит на зміну конфігурації

Результат роботи запиту відображено на рисунку 3.10. Дана відповідь свідчить про успішне збереження конфігурації.

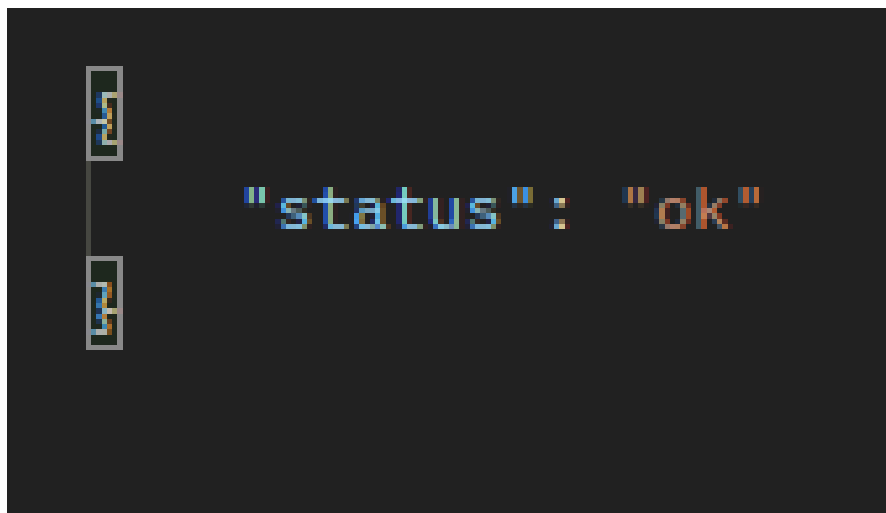


Рисунок 3.10 – Результат виконання запиту

Для перевірки збереження параметрів можна також відкрити сам файл у бекенді. На рисунку 3.11 можна побачити, що збережена конфігурація відповідає той, що була у запиті.

```
backend > static > results > {} config.json > ...
1  {
2    "fitness_threshold": 100,
3    "generations": 100,
4    "mutation_rate": 0.1,
5    "parabola_half_length": 30,
6    "pop_size": 400
7  }
```

Рисунок 3.11 – Перевірка зміни конфігурації у config.json

На рисунку 3.12 показано запит до інтерфейсу, що відповідає за старт алгоритму. За один раз можна вказати декілька сканів, що важливо для автоматизації обробки георадарних даних.

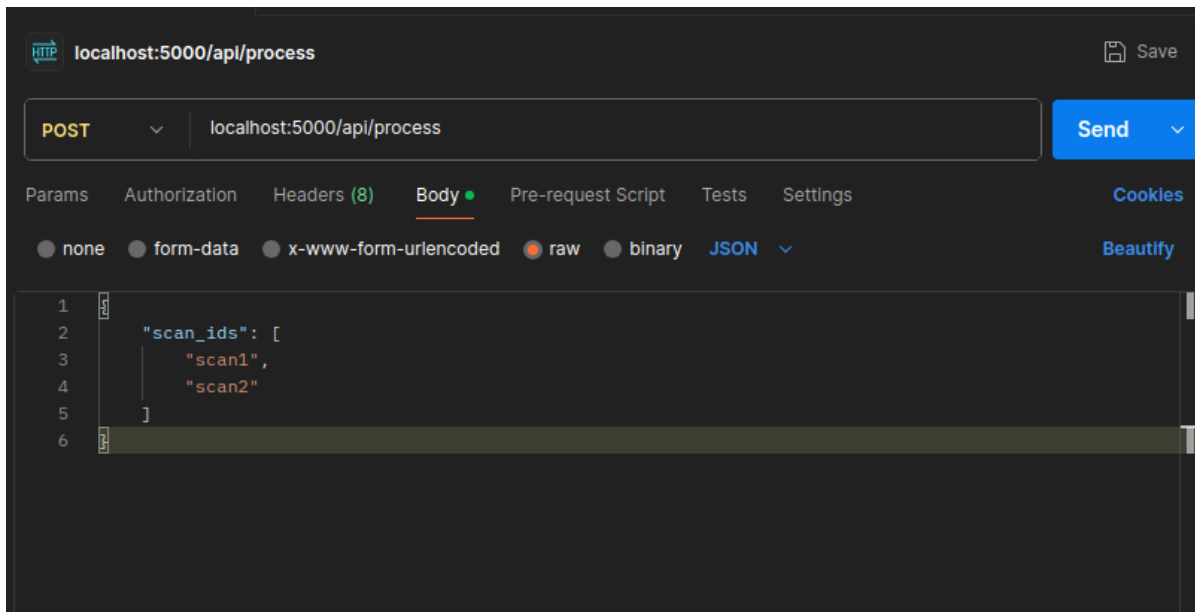


Рисунок 3.12 – Запит на обробку двох сканів

У відповідь бекенд повертає дані, що зображені на рисунку 3.13.

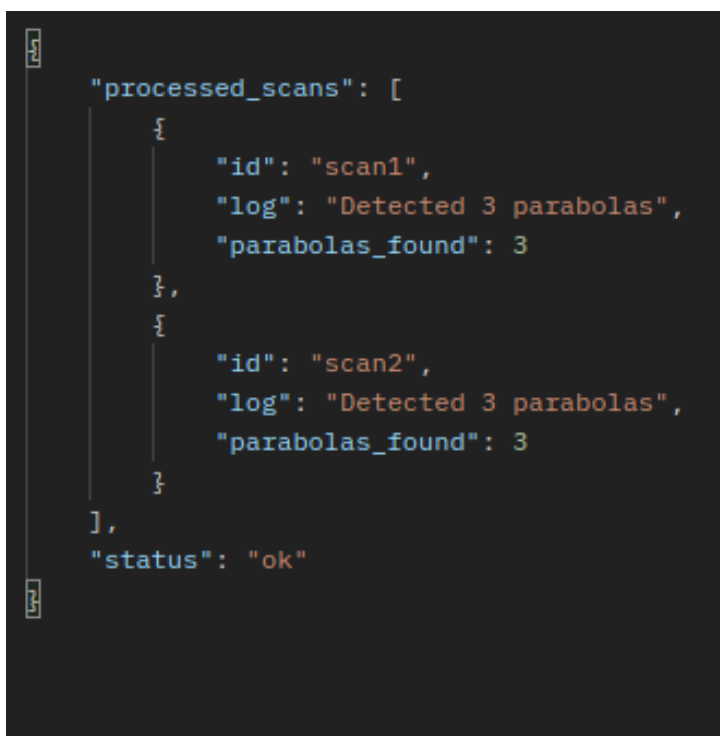


Рисунок 3.13 – Результат виконання запиту

3.2.2 Попередня обробка зображення

Для попередньої обробки зображення необхідно видалити статичне відбиття, відбиття від землі, шуми, також необхідно виділити ознаки парабол для кращої детекції. Після цього виконується виділення регіону інтересу (ROI) та нормалізація зображення за допомогою методу MinMax до діапазону значень -128, 127.

3.2.3 Створення алгоритму

Алгоритмічний модуль є частиною бекенду та розташований у файлі `algorithm.py`. Він включає в себе функції попередньої обробки сканів, перетворення його у `heatmap` зображення за допомогою модуля `matplotlib`, та генетичного алгоритму, налаштованого під пошук параболічних відбиттів.

Генетичний алгоритм було спочатку розроблено і налаштовано на синтетичних даних. Для цього було згенеровано бінарне зображення параболі (рис. 3.14).

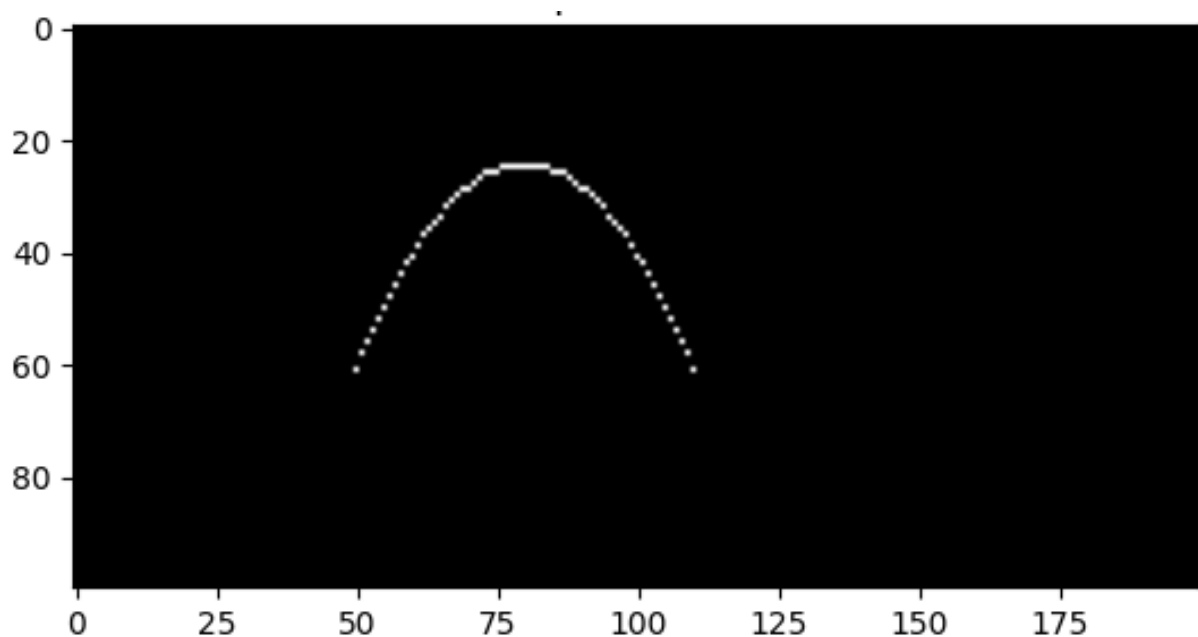


Рисунок 3.14 – Згенерована парабола

На даному синтетичному зображенні немає шумів, тож код алгоритму та функцію пристосованості можна налаштувати достатньо просто. Перший результат роботи алгоритму можна побачити на рисунку 3.15.

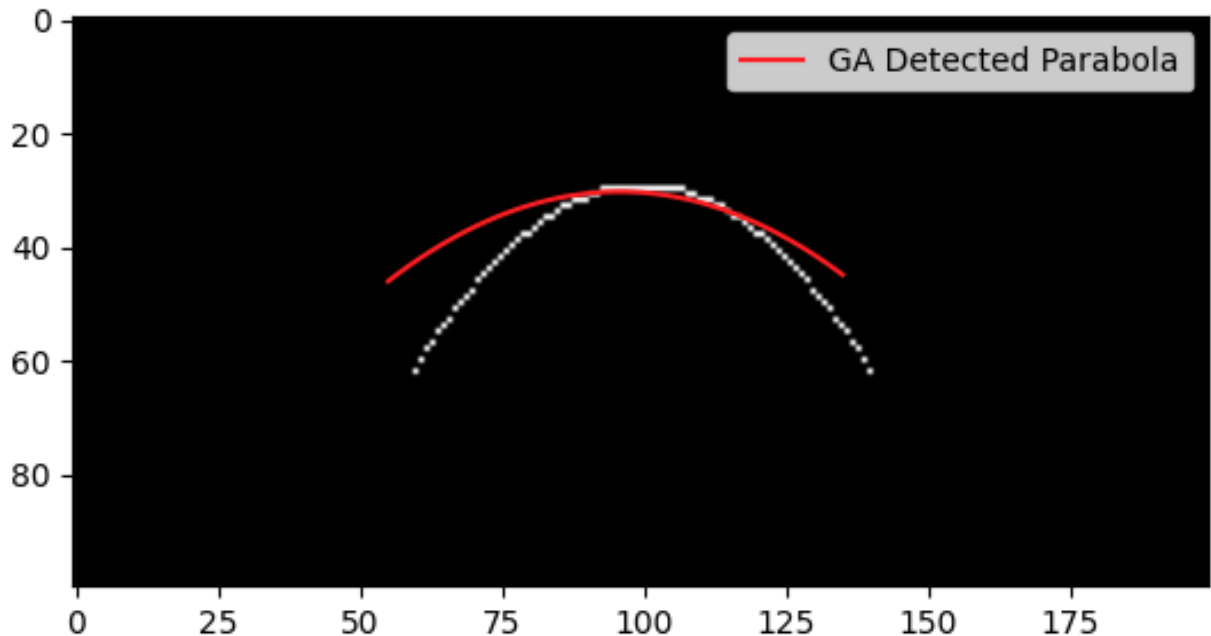


Рисунок 3.15 – Порівняння результату алгоритму та реальної параболи

Бачимо, що найбільш пристосована парабола близька за своїми параметрами до реальної, але, враховуючи, що дане зображення не містить шумів та інших артефактів, можна дійти до висновку, що параметри алгоритму потребують коригування – через недостатню кількість кандидатів в популяції, алгоритм не дійшов до оптимального рішення.

Після збільшення популяції до 50 особин та збільшення поколінь до 100, параметри параболи було знайдено коректно (рис 3.16).

Наступним етапом є додавання шуму до зображення. Для генерації шуму було використано додавання гауссівського шуму до згенерованого зображення синтетичної параболи. Приклад знайденого об'єкта на зашумленому зображенні подано на рис. 3.17. Можна побачити, що результат не є ідеальним, однак це можна виправити, змінивши параметри алгоритму.

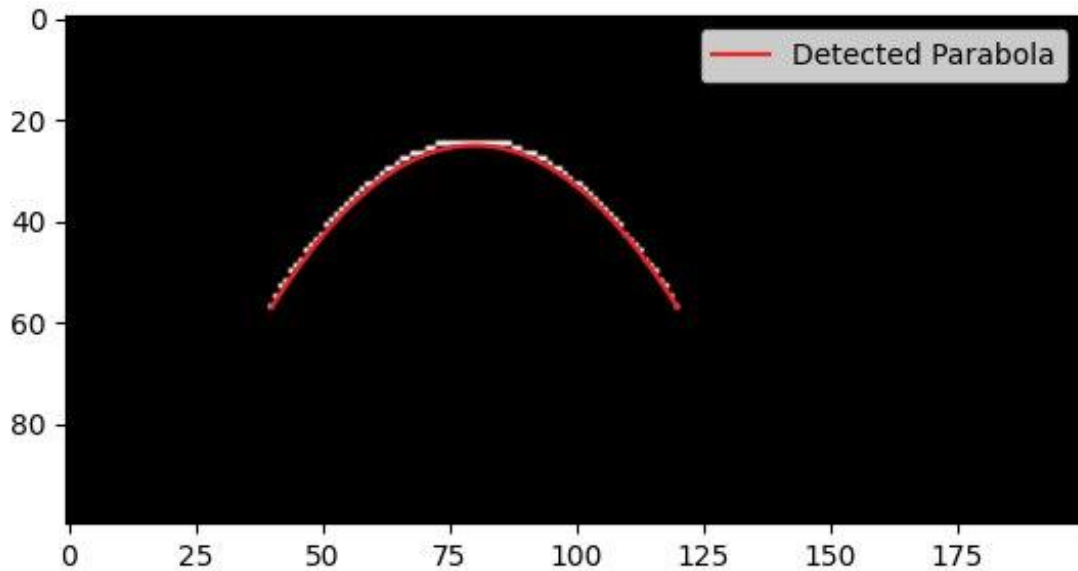


Рисунок 3.16 – Парабола знайдена коректно

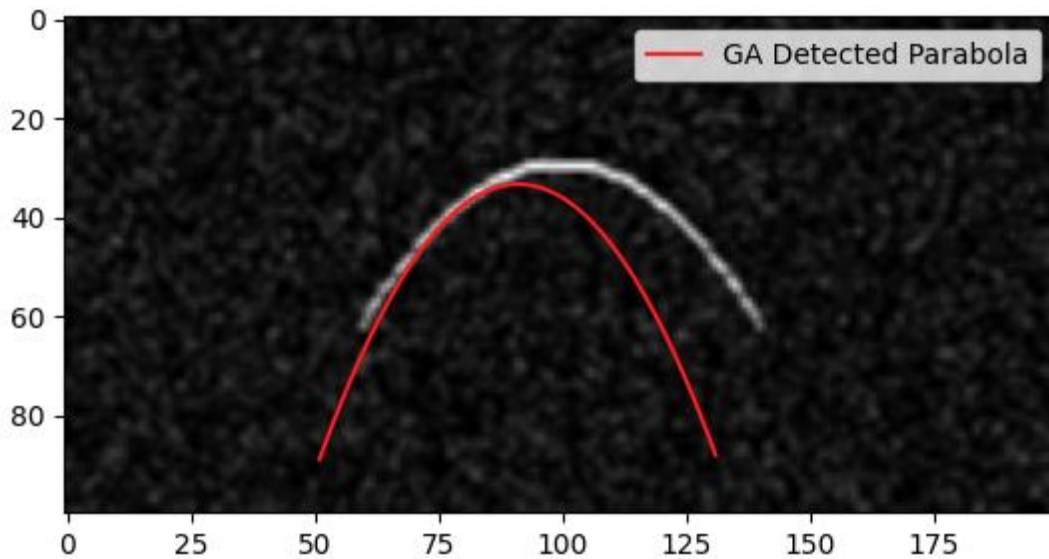


Рисунок 3.17 – Пошук параболі на зашумленому зображенні

Тут складність полягає у тому що функція пристосованості тепер буде оцінювати частини зображення, де параболі немає. Однак місця, де парабола присутня, мають вищу інтенсивність, тож особини, що розташовані на них, мають отримати вище значення функції пристосованості.

Після налаштування алгоритму для пошуку однієї параболі на синтетичному зображенні, необхідно модифікувати його для знаходження декількох парабол (рис 3.18). Після цього алгоритм можна налаштувати для пошуку парабол на реальному скані радару.

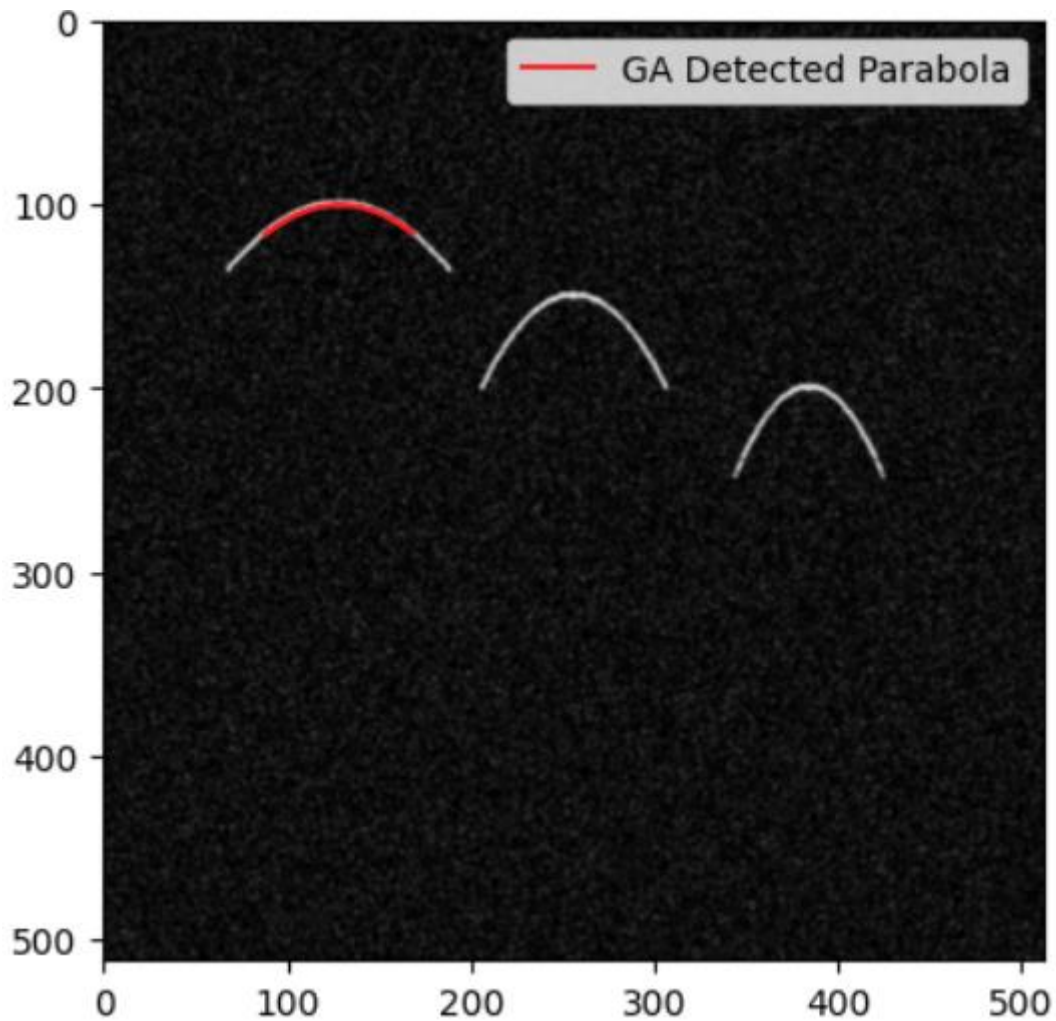


Рисунок 3.18 – Ітерація пошуку декількох парабол на зашумленому зображенні

Для того щоб знаходити параболу на реальному скані, необхідно внести зміни в функцію пристосованості. Треба врахувати діапазон значень, також той факт, що знак амплітуди не є важливим на даному етапі, і необхідно враховувати модуль амплітуди, саме тому було обрано діапазон -128, 127, а не, наприклад, 0, 255.

Після модифікації алгоритму, його було протестовано на реальному скані. Також було імплементовано маскування знайденої параболу та ітеративний виклик алгоритму, поки максимальне значення функції пристосованості не буде менше за певне порогове значення. Результат ітеративного алгоритму на реальному зображенні можна побачити на рис. 3.19.

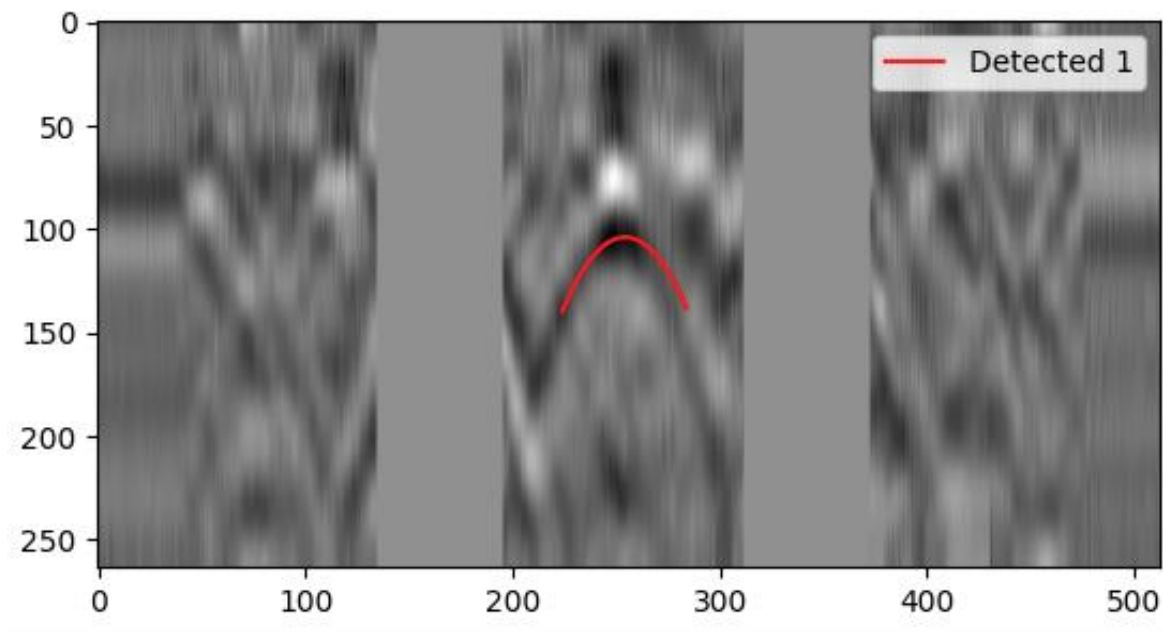


Рисунок 3.19 – Результат пошуку трьох парабол на радарограмі з маскуванням

3.2.4 Створення фронтенду

Клієнтська частина реалізована як односторінковий застосунок на React з використанням Vite для швидкого розгортання й збірки. Vite дозволяє швидко налаштувати проєкт, наприклад, для створення фронтенду необхідно було виконати лише декілька простих команд у терміналі: викликати vite для створення проєкту за шаблоном React та завантажити необхідні для роботи React застосунку модулі.

На рисунку 3.20 відображена структура створеного проєкту з доданими файлами для коректного функціонування фронтенду. Основним файлом тут є src/App.jsx, з нього починається виконання логіки клієнтської частини. У файлі api/scans.js за допомогою бібліотеки axios, налаштовуються функції, що відправляють запити до бекенду. Функції є асинхронними, тобто вони не блокують виконання основного коду, а дозволяють відтермінувати виконання логіки, що використовує ці функції.

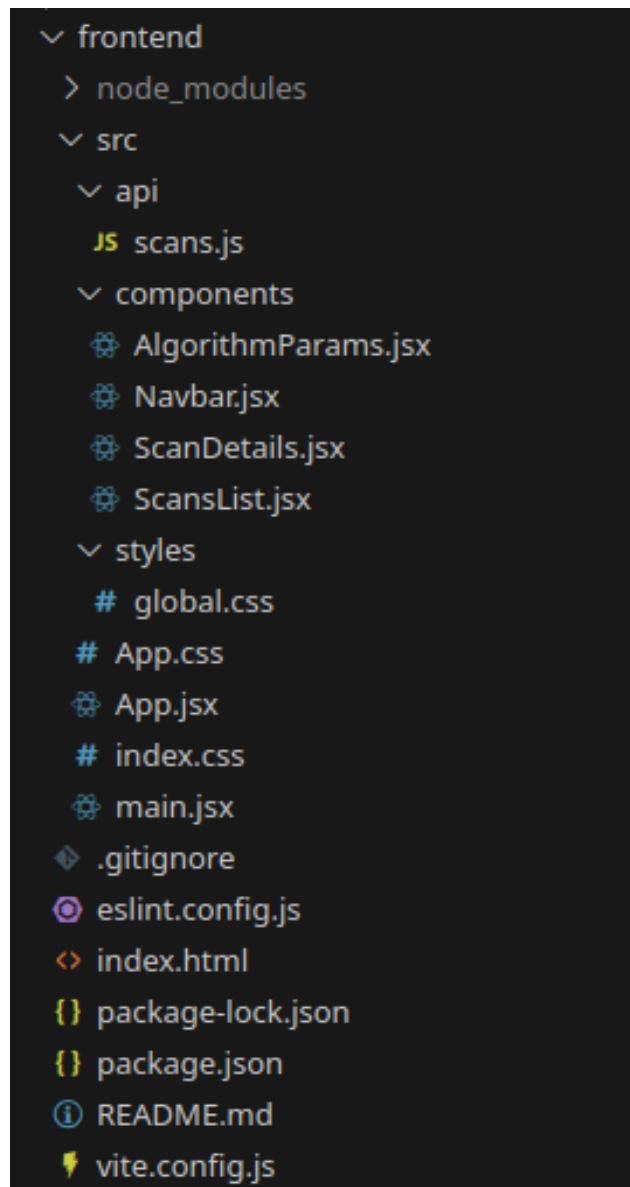


Рисунок 3.20 – Структура фронтенду

Лістинг 3.2 Реалізація функцій запитів до бекенду:

```
import axios from 'axios';  
  
const api = axios.create({  
    baseURL: 'http://localhost:5000/api',  
});  
  
export async function fetchScans() {
```

```
const response = await api.get('/scans');
return response.data;
}

export async function fetchScanDetails(scanId) {
  const response = await api.get(`/scans/${scanId}`);
  return response.data;
}

export async function saveConfig(config) {
  await api.post('/config', config);
}

export async function loadConfig() {
  const resp = await api.get('/config');
  return resp.data;
}

export async function runAlgorithm(scanIds) {
  const response = await api.post('/process',
    { scan_ids:
      scanIds });

  return response.data;
}
```

Головне вікно розбите на два розділи – «Список сканів» і «Параметри алгоритму» – між якими користувач перемикається за допомогою React Router. При відкритті застосунку завантажується список доступних сканів із бекенду, а кожна дія викликає HTTP-запит через Axios. Інтерфейс відображає статуси,

логи та результати у вигляді зображень і таблиць. Вигляд головної сторінки показано на рисунку 3.21.

Лістинг 3.3 Реалізація перемикання розділів сторінки:

```

<BrowserRouter>
  <Navbar />
  <div className=
    »container»>
    <Routes>
      <Route path=»/» element={<Navigate to=»/scans» replace />} />
      <Route path=»/scans» element={<ScansList />} />
      <Route path=»/scans/:scanId» element={<ScanDetails />} />
      <Route path=»/params» element={<AlgorithmParams />} />
    </Routes>
  </div>
</BrowserRouter>

```

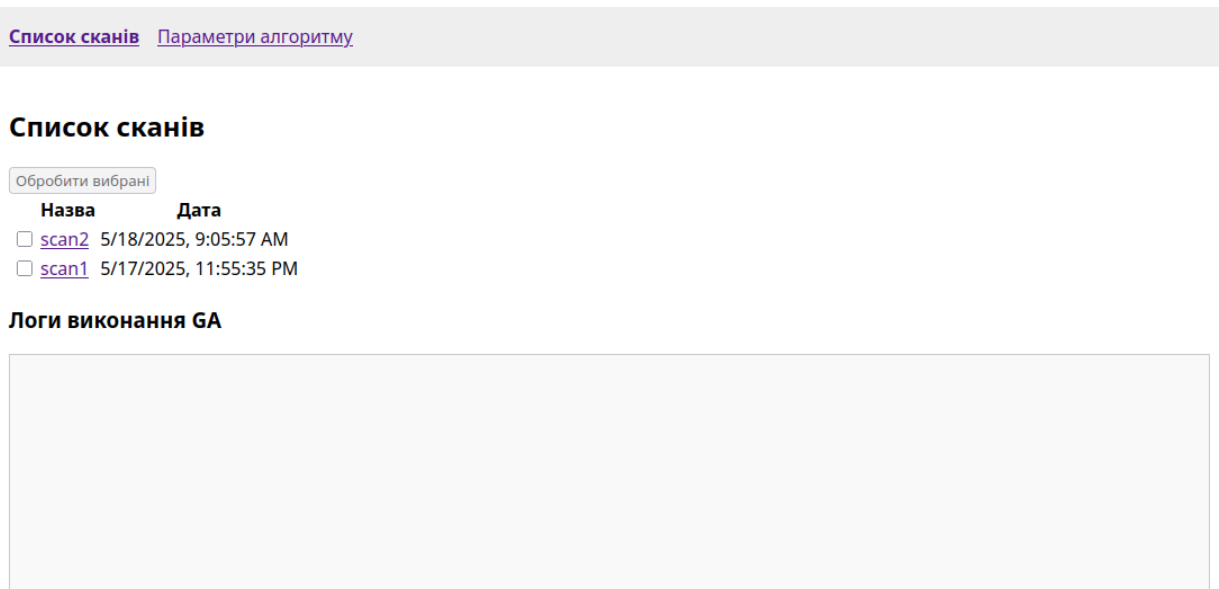


Рисунок 3.21 – Вигляд основної сторінки

Тут оператор може бачити скани, що готові до обробки алгоритмом, вибрати деякі з них, та почати виконання алгоритму за допомогою кнопки «Обробити вибрані», та спостерігати за виконанням через вікно «Логи виконання GA». Кнопка «Обробити вибрані» стає неактивною, якщо жоден скан не вибраний (рис. 3.22). При виборі хоча б одного скану, кнопка стає активною (рис. 3.23).

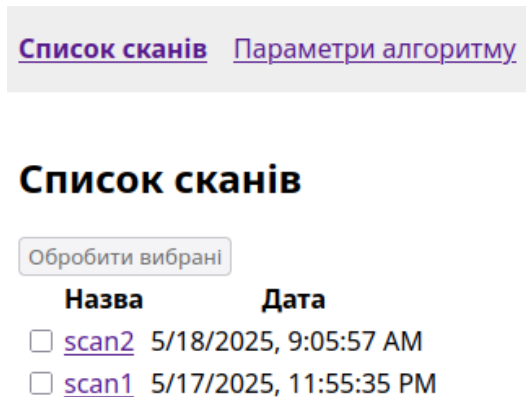


Рисунок 3.22 – Неактивна кнопка «Обробити вибрані»

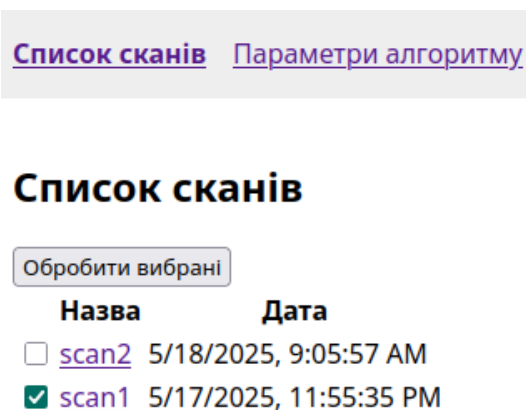


Рисунок 3.23 – Активна кнопка «Обробити вибрані»

Після вибору сканів для обробки, користувач може почати обробку вибраних сканів, натиснувши на кнопку «Обробити вибрані» (рис. 3.24). У полі «Логи виконання GA» у реальному часі відображається журнал повідомлень статусу обробки сканів. Завдяки такому підходу користувач може бачити поточну задачу, яку виконує алгоритм.

[Список сканів](#) [Параметри алгоритму](#)

Список сканів

Обробити вибрані

Назва	Дата
<input type="checkbox"/> scan2	5/18/2025, 9:05:57 AM
<input checked="" type="checkbox"/> scan1	5/17/2025, 11:55:35 PM

Логи виконання GA

```
Begin search for 'scan1'
Generation 0, best fitness: 7744.5
Generation 10, best fitness: 8441.3
Generation 20, best fitness: 8506.4
Generation 30, best fitness: 8511.4
Generation 40, best fitness: 8511.4
Generation 50, best fitness: 8511.4
Generation 60, best fitness: 8511.4
Generation 70, best fitness: 8512.6
Generation 80, best fitness: 8512.6
Generation 90, best fitness: 8519.3
Generation 99, best fitness: 8519.3
```

Рисунок 3.24 – Логи початку роботи алгоритму

У вікні повідомлень можна також побачити, коли алгоритм було завершено (рис. 3.25).

[Список сканів](#) [Параметри алгоритму](#)

Список сканів

Обробити вибрані

Назва	Дата
<input type="checkbox"/> scan2	5/18/2025, 9:05:57 AM
<input checked="" type="checkbox"/> scan1	5/17/2025, 11:55:35 PM

Логи виконання GA

```
Generation 40, best fitness: 1858.3
Generation 50, best fitness: 2055.2
Generation 60, best fitness: 2061.7
Generation 70, best fitness: 2115.1
Generation 80, best fitness: 2115.1
Generation 90, best fitness: 2211.0
Generation 99, best fitness: 2231.2
[91.6283131342547, 74.62118017906216, 0.008533064937135106]
best fitness best_fitness=2231.2
fitness is less than threshold '2500', stop searching
End search for 'scan1'. Found objects: 3
Finish processing scans
```

Рисунок 3.25 – Логи завершення роботи алгоритму

У верхньому кутку знаходиться навігаційна панель, яка дозволяє перейти до налаштувань самого алгоритму (рис. 3.26).

[Список сканів](#) [Параметри алгоритму](#)

Параметри алгоритму

fitness_threshold:

generations:

mutation_rate:

parabola_half_length:

pop_size:

Рисунок 3.26 – Налаштування алгоритму

Для перегляду сканів необхідно натиснути на посилання, що є назвою скану, щоб перейти до детального перегляду. Після цього на сторінці з'явиться декілька зображень радарограм. Перше зображення – це теплова карта (heatmap) необробленого скану (рис. 3.27). Далі показано теплову карту з використанням методів фільтрації, тут користувач може самостійно проаналізувати скан, якщо виникає така потреба (рис. 3.28). Наступне зображення – теплова карта з позначеними знайденими об'єктами (рис. 3.29). Об'єкти позначені параболою, що побудовані на основі знайдених алгоритмом параметрів. Це дозволяє користувачу впевнитись у коректності роботи алгоритму для його доналаштування.

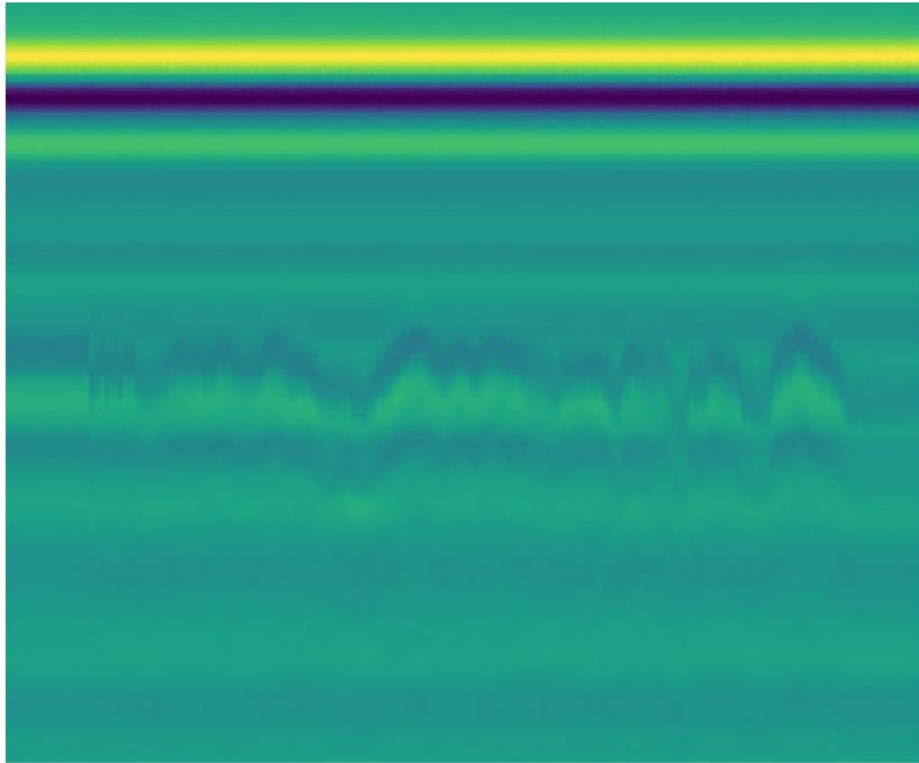
Деталі скану: scan1**Необроблений скан з радару**

Рисунок 3.27 – Зображення необробленого скану

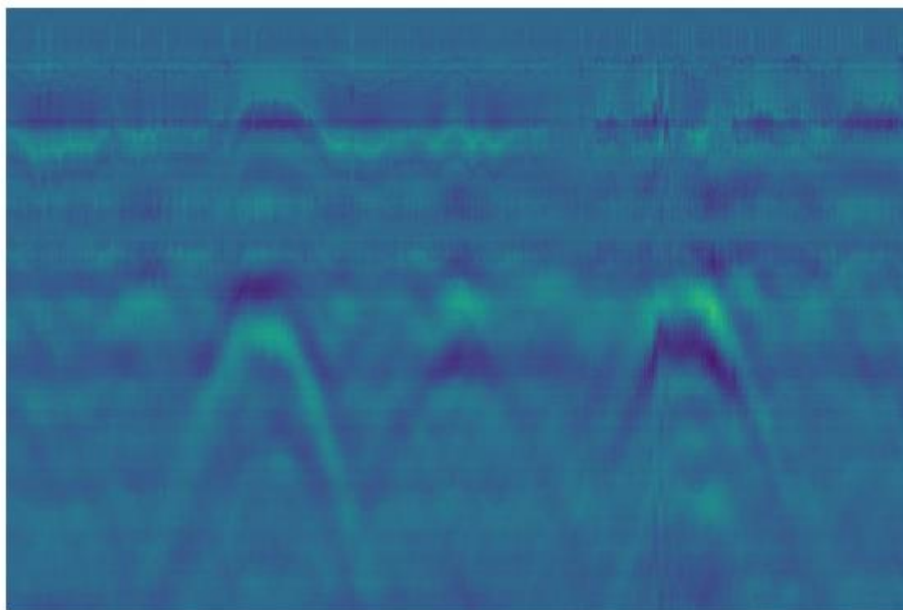
Попередньо оброблений скан з радару

Рисунок 3.28 – Зображення обробленого скану

Скан з позначеними знайденими об'єктами

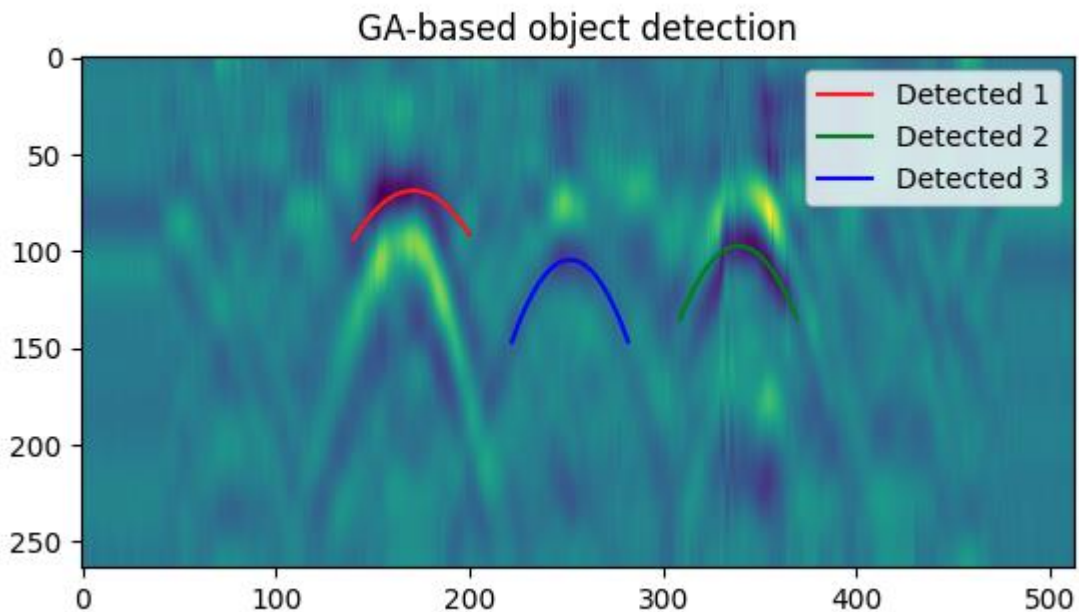


Рисунок 3.29 – Ілюстрація знайдених об'єктів

Використання параболічних ліній замість точкових позначень вершин дає можливість зрозуміти, які частини скану алгоритм оцінив більше та чому. Після скану з позначеними об'єктами розташована таблиця зі знайденими параметрами об'єктів (рис. 3.30).

Знайдені об'єкти

#	x_0	y_0	a	fitness
1	171.0	68.9	0.0265	7997.1
2	339.1	97.5	0.0414	6772.2
3	252.0	104.6	0.0469	2973.1

Рисунок 3.30 – Параметри знайдених на скані об'єктів

Параметри, як такі, не мають особливої цінності для користувача, але вони можуть бути експортовані до інших програм, щоб, наприклад, знаючи GPS-координати радару у момент запису радарограми, вирахувати координати знайдених об'єктів, та позначити їх на онлайн-мапі. Більшість онлайн-мап має зручний інтерфейс для зміни їх вигляду, додавання своїх позначок або маршрутів. Прикладами таких сервісів є Google Maps та OpenStreetMap.

Для автоматичного експорту параметрів знайдених об'єктів можна використати файл з результатами обробки скану (рис. 3.31).

```
backend > static > results > found_objects > {} scan1.json > ...
1  {
2    "filename": "scan1.png",
3    "parabolas": [
4      {
5        "x0": 166.687785064284,
6        "y0": 68.68440678429647,
7        "a": 0.03836439613579714,
8        "fitness": 8423.426229508197
9      },
10     {
11       "x0": 342.92378952758264,
12       "y0": 96.18250522918044,
13       "a": 0.052384997289825055,
14       "fitness": 6921.098360655737
15     },
16     {
17       "x0": 263.06283430080185,
18       "y0": 104.50531190983082,
19       "a": 0.008279905698009255,
20       "fitness": 2802.344262295082
21     }
22   ],
23   "width": 513,
24   "height": 513,
25   "timestamp": "2025-05-19T12:02:34.604739"
26 }
```

Рисунок 3.31 – Збережений файл з результатами роботи алгоритму

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено вебзастосунок, що призначений для виявлення об'єктів на невеликій глибині на основі даних георадарного (GPR) сканування із застосуванням еволюційних алгоритмів.

Розроблене рішення є інтеграцією методів обробки сигналів та елементів штучного інтелекту, що забезпечує високу ефективність у вирішенні задач у сфері георадарного зондування.

Створена система значно спрощує та пришвидшує аналіз польових даних, забезпечуючи автоматичний пошук об'єктів на глибині до декількох десятків сантиметрів.

Модель адаптована для роботи в умовах обмеженої обчислювальної потужності й здатна обробляти дані в реальному часі або у пакетному режимі, що робить її потужним інструментом у польових умовах. Вона є відкритою та безкоштовною, може бути використана на різних платформах та легко масштабуватися під серії однотипних сканів.

Подальші дослідження будуть спрямовані на оптимізацію алгоритмів попередньої обробки та пошуку об'єктів, покращення результатів в умовах високої зашумленості, нерівномірного руху радару, покращення візуального інтерфейсу, обробку даних для радару з кількома антенами-приймачами, та знаходження GPS-координат об'єктів. Також важливим етапом є практичне застосування програми у складі системи пошуку підповерхневих вибухових пристроїв з використанням георадару конфігурації 1 Tx + 4 Rx (одна антена-передавач, чотири антени-приймача), встановленому на наземному роботі Jackal UGV (Clearpath) [6]. Додатково можливе покращення алгоритму для отримання розмірів підземних об'єктів, або інтеграція нейронних мереж для класифікації об'єктів [3].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pryshchenko, O. A., Plakhtii, V., Dumin, O. M., Pochanin, G. P., Ruban, V. P., Capineri, L., & Crawford, F. (2022). Implementation of an artificial intelligence approach to GPR systems for landmine detection. *Remote Sensing*, 14(17), 4421.
2. Ruban, V., Ogurtsova, T., Pochanin, G., Capineri, L., Bossi, L., Bechtel, T., & Crawford, F. (2022, October). Object coordinate determination by the impulse GPR with a Tx+ 4Rx antenna system. In *19th International Conference on Ground Penetrating Radar* (pp. 155-158). Society of Exploration Geophysicists.
3. Dumin, O., Plakhtii, V., Pochanin, G., & Shyrokorad, D. (2020, September). Object classification using artificial neural network processing of data obtained by impulse GPR with 1 Tx+ 4Rx antenna system. In *2020 IEEE Ukrainian Microwave Week (UkrMW)* (pp. 1140-1144). IEEE.
4. Baker, G. S., Jordan, T. E., & Pardy, J. (2007). An introduction to ground penetrating radar (GPR).
5. Pochanin, G. P., Ruban, V. P., Batrakova, A. G., Urdzik, S. N., & Batrakov, D. O. (2014, June). Measuring of thickness of the asphalt pavement with use of GPR. In *2014 15th international radar symposium (IRS)* (pp. 1-4). IEEE.
6. Bossi, L., Falorni, P., Pochanin, G., Ruban, V., Ogurtsova, T., Crawford, F., ... & Capineri, L. (2020, November). Detection of targets from radar tracks of the UWB-GPR “1Tx+ 4Rx” on the mobile platform “UGO 1st”. In *18th International Conference on Ground Penetrating Radar* (pp. 448-451). Society of Exploration Geophysicists.
7. Shafronenko, A., Bodyanskiy, Y. V., & Pliss, I. (2023). Credibilistic Fuzzy Clustering Method Based on Evolutionary Approach of Crazy Wolves in Online Mode. In *CMIS* (pp. 141-150).
8. Shafronenko, A. Y., Bodyanskiy, Y. V., & Holovin, O. O. (2023). Clusterization of data arrays based on the modified gray wolf algorithm. *Radio Electronics, Computer Science, Control*, (1), 73.

9. Shafronenko, A. Y., Kasatkina, N. V., Bodyanskiy, Y. V., & Shafronenko, Y. O. (2023). CREDIBILISTIC ROBUST ONLINE FUZZY CLUSTERING IN DATA STREAM MINING TASKS. *Radio Electronics, Computer Science, Control*, (3), 97-97.
10. Shafronenko, A., Bodyanskiy, Y., Pliss, I., & Patlan, K. (2019, June). Fuzzy Clusterization of Distorted by Missing Observations Data Sets Using Evolutionary Optimization. In *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 217-220). IEEE.
11. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень.
12. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
13. Bodyanskiy, Y. V., Shafronenko, A., & Rudenko, D. (2019). Online Neuro Fuzzy Clustering of Data with Omissions and Outliers based on Completion Strategy. In *CMIS* (pp. 18-27).
14. Shafronenko, A., & Bodyanskiy, Y. (2019). Online algorithm for possibilistic fuzzy clustering based on evolutionary cat swarm optimization. *Science and Education a New Dimension. Natural and Technical Sciences*, 193, 86-88.
15. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Identification of visual objects by the search request.
16. Zhou, M., Cui, M., Xu, D., Zhu, S., Zhao, Z., & Abusorrah, A. (2024). Evolutionary optimization methods for high-dimensional expensive problems: A survey. *IEEE/CAA Journal of Automatica Sinica*, 11(5), 1092-1105.
17. Gorokhovatskyi, V. O., & Gadetska, S. V. (2019). Determination of relevance of visual object images by application of statistical analysis of regarding fragment representation of their descriptions. *Telecommunications and Radio Engineering*, 78(3).

18. Gorokhovatskiy, V. A., & Zamula, A. A. (2016). Employment of Intelligent Technologies in Multiparametric Control Systems. *Telecommunications and Radio Engineering*, 75(19).

19. Gorokhovatsky, V. O., & Solodchenko, K. G. (2018). Застосування апарату аналізу та оброблення бітових даних у методах класифікації зображень за множиною ключових точок. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 2(48), 63-67.

20. Гороховатський, В. О., Творошенко, І. С., & Сидоренко, Д. (2021). Класифікація зображень із використанням кластерного подання.

21. Гороховатський, В. О., Гадецька, С. В., Стяглик, Н. І., & Власенко, Н. В. (2020). Класифікація зображень на підставі ансамблю статистичних розподілів за класами еталонів для компонентів структурного опису.

22. Гороховатський, В. О., Пупченко, Д. В., & Солодченко, К. Г. (2018). Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення.

23. Gadetska, S., Gorokhovatskyi, V., Stiahlyk, N., & Vlasenko, N. (2022). Aggregate parametric representation of image structural description in statistical classification methods.

24. Gadetska, S. V., Gorokhovatskyi, V. O., Stiahlyk, N. I., & Vlasenko, N. V. (2021). Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. *Radio Electronics, Computer Science, Control*, (4), 58-68.

25. Gorokhovatsky, V. A., Putyatin, Y. P., & Stolyarov, V. S. (2017). Research of effectiveness of structural image classification methods using cluster data model. *Radio Electronics Computer Science Control*, (3), 78-85.

26. Gorokhovatskyi, V. O. L. O. D. Y. M. Y. R., Gadetska, S. V. I. T. L. A. N. A., & Stiahlyk, N. A. T. A. L. I. A. (2023). Accelerating image classification based on a model for estimating descriptor-to-class distance.

27. Gorokhovatskyi, V., & Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. *Advanced Information Systems*, 5(4), 10-16.

28. Zhang, X., Nilot, E., Feng, X., Ren, Q., & Zhang, Z. (2018). IMF-slices for GPR data processing using variational mode decomposition method. *Remote Sensing*, 10(3), 476.

29. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Handwritten character recognition models based on convolutional neural networks.

30. Tvoroshenko, I. S., & Gorokhovatskyi, V. O. (2019). Intelligent classification of biophysical system states using fuzzy interval logic. *Telecommunications and Radio Engineering*, 78(14).