

**Шифр:** ST6KAI113AS OGP2025AIC

**Тема:** Оцінювання площ областей антропогенної діяльності  
на основі даних повітряного спостереження

## ЗМІСТ

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| ВСТУП.....                                                                                           | 3  |
| <br>РОЗДІЛ 1. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ, НЕОБХІДНІ ДЛЯ<br>СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 6  |
| 1.1. Застосування нейронних мереж до вирішення задачі класифікації.....                              | 6  |
| 1.2. Референц-еліпсоїди та їх основні параметри.....                                                 | 9  |
| 1.3. Забезпечення двостороннього переходу між системами координат.....                               | 10 |
| 1.4. Файли світової прив'язки.....                                                                   | 12 |
| 1.5. Базові поняття сферичної тригонометрії для розв'язування сферичних<br>трикутників.....          | 13 |
| 1.6. Тріангуляція Делоне.....                                                                        | 15 |
| 1.7. Формат XML та GPX для зберігання структурованої інформації.....                                 | 15 |
| <br>РОЗДІЛ 2. ОПИС ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕАЛІЗОВАНОГО<br>ПРОГРАМНОГО КОМПЛЕКСУ.....             | 17 |
| 2.1. Навчання нейромережі.....                                                                       | 18 |
| 2.2. Реалізація файлів світової прив'язки.....                                                       | 21 |
| 2.2.1. Ілюстрований опис ручного створення.....                                                      | 21 |
| 2.2.2. Повний приклад програмного нанесення треку.....                                               | 22 |
| 2.2.3. Нанесення треків з побутових пристроїв на цифрові карти.....                                  | 23 |
| 2.3. Опис методів підрахунку площ.....                                                               | 24 |
| <br>ВИСНОВКИ.....                                                                                    | 27 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....                                                                        | 28 |

## ВСТУП

**Актуальність.** Розуміння географічних положень та взаємозв'язків є дуже важливим для повсякдення. Адресна доставка товарів, транспортування між містами, призначення місця зустрічі втілюються при наявності точних координат шуканого об'єкту. Важливість таких аспектів призвела до активного розвитку сфери вищої геодезії, котра призначена для формалізації установлених понять та термінів.

При вирішенні складноформалізованих задач, направлених на аспект об'єднання вище названих геодезичних сфер, в умовах невизначеності, є нейромережа. Вона допомагає виявляти проблемні аспекти, котрі складно охопити навіть висококваліфікованим спеціалістам з подальшим їх вирішенням. Багато із застосовуваних нині нейромереж можуть вирішувати поставлену задачу лише після навчання, із запропонованим їй архітектором набором прикладів, котрі вона має доволі чітко адаптувати під реальні умови.

В ході аналізу тематичної літератури не було знайдено універсального методу для підрахунку площ багатокутників на еліпсоїді.

**Об'єкт дослідження:** процес оцінювання розміру областей антропогенної діяльності без врахування рельєфу.

**Предмет:** метод штучного інтелекту для класифікації фотоматеріалів, триангуляція, як науковий інтерес обчислювальної графіки, процедури комп'ютерної графіки, котрі є суміжними до геодезичних обрахунків площ полігонів, а також засоби обчислювального інтелекту.

**Мета:** розробити інформаційну технологію для обчислення площі багатокутників, на основі відгуку нейромережевого класифікатора щодо області антропогенної діяльності, алгоритмами вищої геодезії та комп'ютерної графіки, на прикладі зруйнованих будинків, без врахування рельєфу місцевості.

### ***Постановка завдання:***

1. Обрахувати площу зруйнованих будинків за допомогою нейромережевого класифікатора, як результат області антропогенної діяльності на фотоматеріалах повітряного спостереження.
2. Для покращення результатів класифікації, донавчити преднавчену нейромережеву модель ResNet50V2, шляхом додавання допоміжних прикладів, кожен з яких належить до одного з чотирьох обраних класів (“DAMAGED\_BUILDINGS”, “BUILDINGS”, “RESIDENTIAL\_BUILDINGS” та “ROADS”).
3. Застосувати принцип ковзного вікна при аналізі вхідного зображення, із подальшим їх об'єднанням у багатокутники. Такі дії призведуть до отримання меж області антропогенної діяльності в пікселях.
4. Забезпечити обмін та збереження згенерованих даних в XML-форматі, як для піксельних координат вершин, так і для геодезичних.
5. Підготувати чотири опорні точки та здійснити накладання зображення на цифрову карту сервісу Google Maps, отримати координати пікселів й меж області антропогенної діяльності в градусах WGS84.
6. Використати три методи обрахунку площ полігону, координати вершин якого задані в системі WGS84: триангуляція на площині опісля застосування формул проєкції Гаусса-Крюгера, точки для якої відображено на картах СК42 різних масштабів (1:100000, 1:200000, 1:500000); триангуляція на еліпсоїді; безпосереднє вичислення площі багатокутника на еліпсоїді по алгоритму GisLab.
7. Порівняти швидкість обрахунку площ, які здатні надати обрані методи.
8. Реалізувати додаток методами з графічним інтерфейсом із забезпеченням маніпулювання world-файлами, треками маршрутів, використанням явного переходу до плоских прямокутних координат СК42 для вершин полігонів та підрахунок площ, раніше знайдених полігонів.

**Методи дослідження:** вивчення відповідної тематичної літератури з відкритих джерел, аналіз та угруповання отриманої інформації, обчислення та математичне обґрунтування отриманих результатів.

**Практична значимість:** представлений у проєкті підхід може застосовуватись на державних та комерційних підприємствах для фінансової оцінки збитків від пошкодження посівних полів, житлових масивів, автомагістралей та інших об'єктів інфраструктури, з залученням необхідних впроваджень для їх усунення та подальшої реконструкції.

**Використані інструменти:** мови програмування Python та C#, середовище розробки Visual Studio, текстовий редактор Notepad++, хмарне середовище Google Colab, фреймворки TensorFlow, OpenCV та Shapely.

**Особистісний внесок:** програмне забезпечення для проєкту здебільшого розроблено власноруч, також були задіяні функціонали, котрі вже мають готові програмні рішення. Анотовано фотографії для донавчання нейромережі, закодований алгоритм про зміщення ковзного вікна, написана функція обробки файлів у форматі GPX. Написані програми для: алгоритму Боуера-Вотсона для тріангуляції Делоне, перевірки приналежності точки полігону. Створено код для переведення координат із однієї системи в іншу, спроектовано повний цикл для проведення маніпуляцій із world-файлами та можливості використання формул знаходження площ трикутників на площині та на сфері. Базуючись на принципи reverse engineering у C#, переписано метод знаходження площі сферичного полігону з середини обраної бібліотеки.

## **РОЗДІЛ 1. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ, НЕОБХІДНІ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Спочатку розглянемо кілька визначень, пов'язаних з темою роботи.

Оцінка — це процес пошуку величини-апроксимації, яка може бути використана з певною метою [1].

Антропогенний вплив — в перекладі «породжений людиною». До антропогенних відносяться фактори, які зобов'язані прямому впливу від діяльності людини [2].

Геодезія — це наука, котра вивчає форми і розміри Землі, її зображення на планах і картах, точні вимірювання на місцевості [3].

### **1.1. Застосування нейронних мереж до вирішення задачі класифікації**

Нейромережа (англ. neural nets) — система для обчислень, ідея внутрішньої будови якої запозичена з біологічних нейронних мереж, які складають головний мозок живих створінь [4].

Основною характеристикою всього машинного навчання, в більшості випадків, можна вважати процес тренування, під час якого комп'ютерна програма отримує достатньо велику кількість даних та набір інструкцій. Після цього етапу на програму покладається задача пошуку закономірностей у представлених їй даних, щоб досягти поставлених цілей [5].

Математиком Аланом Тьюрингом у 1950 році був запропонований Тест Тьюринга - це метод визначення якості роботи штучного інтелекту (ШІ). Тест перевіряє, чи може машина продемонструвати людський інтелект. Такий тест вважатиметься пройденим тільки тоді, коли машина після вступу в розмову з людиною, не буде розпізнана як звичайна комп'ютерна програма [6].

Першим, хто дав визначення «машинному навчанню» став Артур Самуель (Arthur Samuel) в 1952 році (область досліджень розробки машин, які попередньо не є запрограмованими) [7].

У контексті прикладних алгоритмів машинного навчання (МН), для оцінки якості їх роботи, часто вводять такі поняття, як «похибка під час навчання» (training loss) та «втрата при тестуванні» (validation loss). Наведемо пояснення до мети використання кожної з них, в розрізі задачі класифікації:

- 1) Втрати при навчанні обраховуються в ході реалізації процесу навчання та застосовуються при оновленні параметрів моделі для досягнення максимально можливої в поточних умовах їх мінімізації.
- 2) Похибка узагальнення – це міра того, наскільки добре модель МН працює на окремому наборі даних, котрий явно не використовувався при навчанні. Подібно до похибки навчання, продуктивність моделі на перевірочному наборі можна перевіряти після кожної епохи [8].

Метод зворотнього поширення похибки (від англ. backpropagation) — це основоположний алгоритм для оцінки вектора градієнту від функції похибки, під час контрольованого навчання штучних нейронних мереж, із використанням градієнтного спуску. Саме обчислення градієнту відбувається по відношенню до ваг (weights) та зсувів (biases) нейронної мережі. Назва методу визначає принцип його функціонування та реалізації, тобто обчислення градієнта відбувається у зворотному напрямку через мережу, при цьому градієнт по вагам (та, опціонально, зсувам) останнього шару обчислюється першим, а по вагам першого - останнім. Часткові обчислення градієнта з одного шару повторно використовуються для обчислення градієнта для попереднього шару, що дозволяє певним чином зменшити кількість необхідних обчислювальних операцій, що в поєднанні з сучасними спеціалізованими графічними процесорами надають ще більшу продуктивність. Цей метод було винайдено в 1970-х роках як загальний метод оптимізації для виконання автоматичного диференціювання складних вкладених функцій. Однак тільки в 1986 році, коли автори Румельхарт, Хінтон і Вільямс опублікували статтю під назвою «Навчання представлень шляхом зворотного поширення помилок», важливість алгоритму була по-справжньому оцінена спільнотою машинного навчання [9].

Навчання багатьох нейромереж базується на математичних поняттях.

Градiєнтний спуск — це алгоритм оптимізації, який частіше за все використовується для задач безумовної оптимізації або ж для навчання моделей машинного навчання та нейронних мереж.

Чи не єдиним вхідним значенням алгоритму є початкова точка — це довільна точка для оцінки продуктивності. Метод відноситься до методів першого порядку, тому що використовує значення функції та її частинних похідних першого порядку. Швидкість навчання (розмір кроку або альфа) — це розмір кроків, які потрібно зробити в бік потенційного екстремуму. Частіше за все його значення відносно мале, й оцінюється та оновлюється з врахуванням набуваних значень функції вартості, проте загальних вимог до її вибору немає.

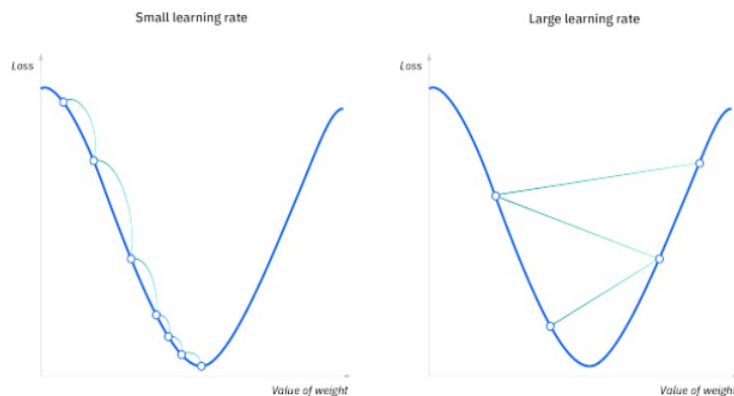


Рис. 1.1. Візуальна різниця між малим та великим кроками навчання

Функція вартості (або втрат) вимірює різницю між фактичним і прогнозованим результатом. Це дозволяє покращити ефективність моделі машинного навчання, надаючи зворотній зв'язок, щоб вона могла коригувати параметри та, відповідно, мінімізувати помилку [10].

Стохастичний градієнтний спуск — широкоживана варіація градієнтного спуску, котрий приміняється під час оптимізації моделей машинного навчання. Замість використання всього набору даних, цей метод для обчислення градієнта та оновлення параметрів моделі пропонує на кожній ітерації обирати лише один випадковий приклад навчання (або невеликий пакет спостережень) [11].



При використанні нейромереж (по типу класифікаторів) навчених на даних малої розмірності, застосовують техніку “ковзного вікна” — це алгоритмічний підхід, який використовується в інформатиці та обробці сигналів. Він передбачає вибір підмножини фіксованого розміру або «вікна» з більшого набору даних і поетапне переміщення цього вікна через набір даних. Вікно ковзає по даним, як правило, по одному елементу за раз, і виконує деякі операції над елементами у вікні на кожному кроці. Використання “ковзного вікна” також можливе у наступних задачах:

- стиснення даних;
- обробка зображень та сигналів;
- передачі даних через мережеві протоколи [12].

## 1.2. Референц-еліпсоїди та їх основні параметри

Референц-еліпсоїд приймається для обробки геодезичних вимірів і встановлення системи геодезичних координат. Існує приблизно 20 референц-еліпсоїдів, приведемо два найбільш поширених із відповідними параметрами:

*Таблиця 1*

Параметри двох обраних сфероїдів

| Назва параметру                        | Еліпсоїд WGS 84   | Еліпсоїд Красовського (СК42) |
|----------------------------------------|-------------------|------------------------------|
| Екваторіальний радіус $a$              | 6 378 137 м       | 6 378 245 м                  |
| Полярний радіус $b$                    | 6 356 752.3142 м  | 6 356 863.01877 м            |
| Полярне стиснення $\alpha$             | 1 / 298.257223563 | 1 / 298.3                    |
| Квадрат першого ексцентриситету $e^2$  | 0.00669437999014  | 0.006693421623               |
| Квадрат другого ексцентриситету $e'^2$ | 0.00673949674228  | 0.006738525415               |

Важливо пам'ятати, що проекція еліпсоїда СК42 не зберігає площі, а зберігає лише відстані, а проекція еліпсоїда WGS84 — зберігає кут. За своєю природою референц-еліпсоїди відносять до поверхні другого порядку.

### 1.3. Забезпечення двостороннього переходу між системами координат

Двосторонній перехід означає можливість проведення розрахунків відносно будь-якої з двох систем координат (як приклад, між WGS84 та СК42). Для таких систем використовують формули перетворення Молоденського [13]:

$$\Delta B = \frac{1}{M_1 + H_1} [-\Delta X \sin(B_1) \cos(L_1) - \Delta Y \sin(B_1) \sin(L_1) + \Delta Z \cos(B_1) + \\ + \frac{a_2 - a_1}{a_1} N_1 e_1^2 \sin(B_1) \cos(B_1) + (\alpha_2 - \alpha_1) (M_1 \frac{a_1}{b_1} + N_1 \frac{b_1}{a_1}) \sin(B_1) \cos(B_1)], \quad (1.1)$$

$$\Delta L = \frac{1}{M_1 + H_1} (-\Delta X \sin(L_1) + \Delta Y \cos(L_1) \sec(B_1)), \quad (1.2)$$

$$\Delta H = \Delta X \cos(B_1) \cos(L_1) + \Delta Y \cos(B_1) \sin(L_1) + \Delta Z \sin(B_1) - \\ - \frac{a_1(a_2 - a_1)}{N_1} + (\alpha_2 - \alpha_1) \frac{b_1}{a_1} N_1 \sin^2(B_1), \quad (1.3)$$

де  $B$  — широта;

$L$  — довгота;

$N_1$  — радіус кривизни першого вертикалу;

$a_1, b_1$  та  $a_2, b_2$  — значення довжин вісей еліпсоїдів;

$\alpha_1, \alpha_2$  — полярні стиснення;

$M_1$  — радіус кривизни меридіани;

$\Delta B, \Delta L, \Delta H$  — поправка до координат  $B_1, L_1, H_1$ .

Також використовують формули для обчислення плоских прямокутних координат  $x, y$  за геодезичними  $B, l$  [14]:

$$\begin{cases} x = X + a_2 l^2 + a_4 l^4 + a_6 l^6 + \dots, \\ y = b_1 l + b_3 l^3 + b_5 l^5 + b_7 l^7 + \dots, \end{cases} \quad (1.4)$$

де 
$$X = \int_0^B M dB.$$

Коефіцієнти рядів у формулі 1.4 знаходять як вирази від широти:

$$\begin{aligned}
b_1 &= N \cos(B), \\
b_3 &= \frac{1}{6} N \cos^3(B) (1 - \operatorname{tg}^2(B) + e'^2 \cos^2(B)), \\
b_5 &= \frac{1}{120} N \cos^5(B) (5 - 18 \operatorname{tg}^2(B) + \operatorname{tg}^4(B) + 14 e'^2 \cos^2(B) - \\
&\quad - 58 \operatorname{tg}^2(B) e'^2 \cos^2(B)), \\
b_7 &= \frac{1}{5040} N \cos^7(B) (61 - 479 \operatorname{tg}^2(B) + 179 \operatorname{tg}^4(B) - \operatorname{tg}^6(B)),
\end{aligned} \tag{1.5}$$

$$\begin{aligned}
a_2 &= \frac{1}{2} N \cos(B) \sin(B), \\
a_4 &= \frac{1}{24} N \cos^3(B) \sin(B) (5 - \operatorname{tg}^2(B) + 9 e'^2 \cos^2(B) + 4 e'^4 \cos^4(B)), \\
a_6 &= \frac{1}{720} N \cos^5(B) \sin(B) (61 - 58 \operatorname{tg}^2(B) + \operatorname{tg}^4(B) + 270 e'^2 \cos^2(B) - \\
&\quad - 330 \operatorname{tg}^2(B) e'^2 \cos^2(B)).
\end{aligned} \tag{1.6}$$

Довжина дуги меридіана:

$$X = A_0 B - A_2 \sin(2B) + A_4 \sin(4B) - A_6 \sin(6B) + \dots, \tag{1.7}$$

Допоміжні значення з формули 1.7 для еліпсоїда Красовського виглядають так:

$$\begin{aligned}
A_0 &= 6367558.4883, \\
A_2 &= 16036.4734, \\
A_4 &= 16.8263, \\
A_6 &= 0.0216.
\end{aligned} \tag{1.8}$$

Наступними є формули для обчислення геодезичних координат  $B, l$  за плоскими прямокутними  $x, y$  [15]

$$\begin{aligned}
B &= B_0 + \Delta B, \\
L &= \frac{6(n-0.5)}{57.29577951} + l,
\end{aligned} \tag{1.9}$$

де  $n = [y * 10^{-6}]$  - номер градусної зони.

Для зручності виведення результуючих співвідношень, вводять допоміжні

$$\text{змінні виду } \beta = \frac{x}{6367558.4968} \text{ та } z_0 = \frac{y - (10n + 5) * 10^5}{6378245} \cos(B_0).$$

Фінальні формули:

$$\begin{aligned}
B_0 &= \beta + \sin(2\beta) (0.00252588685 - 0.00001491860 \sin^2(\beta) + \\
&\quad + 0.00000011904 \sin^4(\beta)),
\end{aligned} \tag{1.10}$$

$$\Delta B = -z_0^2 \sin(2B_0) (0.251684631 - 0.003369263 \sin^2(B_0) + 0.000011276 \sin^4(B_0) - z_0^2 (0.10500614 - 0.04559916 \sin^2(B_0) + 0.00228901 \sin^4(B_0) - 0.00002987 \sin^6(B_0) - z_0^2 (0.042858 - 0.025318 \sin^2(B_0) + 0.014346 \sin^4(B_0) - 0.001264 \sin^6(B_0) - z_0^2 (0.01672 - 0.00630 \sin^2(B_0) + 0.01188 \sin^4(B_0) - 0.00328 \sin^6(B_0))))), \quad (1.11)$$

$$l = z_0 (1 - 0.0033467108 \sin^2(B_0) - 0.0000056002 \sin^4(B_0) - 0.0000000187 \sin^6(B_0) - z_0^2 (0.16778975 + 0.16273586 \sin^2(B_0) - 0.00052490 \sin^4(B_0) - 0.00000846 \sin^6(B_0) - z_0^2 (0.0420025 + 0.1487407 \sin^2(B_0) + 0.0059420 \sin^4(B_0) - 0.0000150 \sin^6(B_0) - z_0^2 (0.01225 + 0.09477 \sin^2(B_0) + 0.03282 \sin^4(B_0) - 0.00034 \sin^6(B_0) - z_0^2 (0.0038 + 0.0524 \sin^2(B_0) + 0.0482 \sin^4(B_0) + 0.0032 \sin^6(B_0)))))). \quad (1.12)$$

#### 1.4. Файли світової прив'язки

Файли світової прив'язки відкривають можливість переходу від одного простору в інший, шляхом лінійного перетворення. Кожен world-файл складається з шести значень, пояснення до яких наведемо в таблиці нижче [16]:

Таблиця 2

Зміст параметрів world-файлу

| Позначення | Можливе значення   | Пояснення значення                                                                   |
|------------|--------------------|--------------------------------------------------------------------------------------|
| A          | 16.8911816889906   | розмір пікселя в одиницях карти вздовж напрямку вісі Oх                              |
| D          | -0.607535968648027 | коефіцієнти обертання вздовж обох вісей (D – Oх, B – Oу)                             |
| B          | -0.61594477098508  |                                                                                      |
| E          | -16.9037948924953  | розмір пікселя в одиницях карти в напрямку вісі ординат; завжди менше нуля           |
| C          | 6286247.45633729   | множники для паралельного перенесення; точка (C, F) – центр верхнього лівого пікселя |
| F          | 5621774.04917468   |                                                                                      |

В таблиці наведено приклади значень для кожного з шести параметрів, які можуть відрізнятись, тому, варто навести можливу процедуру наповнення файлів світової прив'язки у вигляді метод на основі вирішення двох систем лінійних алгебраїчних рівнянь (СЛАР):

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}$$

Матриця коефіцієнтів буде однаковою для обох СЛАР. Праві частини:  $(x'_1 \ x'_2 \ x'_3)^T$  (знайдемо А, В та С) й  $(y'_1 \ y'_2 \ y'_3)^T$  (пошук D, E та F).

Файли світової прив'язки переводять пікселі в обрані географічні координати та навпаки. Для визначення географічного положення точки на основі пікселів застосовують формули переведення піксельних координат в прямокутні (проміжні етапи пропущено):

$$\begin{cases} x = \frac{x' - B * y - C}{A}, \\ y = \frac{(y' - F) * A + (C - x') * D}{AE - BD}. \end{cases} \quad (1.16)$$

### 1.5. Базові поняття сферичної тригонометрії для розв'язування сферичних трикутників

Сферична тригонометрія - розділ геометрії де визначальними об'єктами виступають багатокутники на сфері (здебільшого це трикутники) та взаємозв'язки між їх складовими (сторонами та кутами).

Двокутником (lune) називається частина простору сфери, що розташована між двома сферичними півколами (great semicircles).

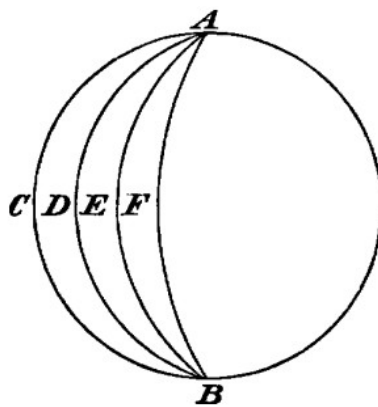


Рис. 1.2. Представлення двокутників на сфері

Наприклад, на рисунку 1.2, ACBDA та ADBEA — двокутники. Припускаючи, що кожна з них має один і той самий кут при А, отримаємо важливу властивість цих об'єктів: пара двокутників рівна при умові рівності їх кутів. Якщо провести три двокутникитак, щоб вони попарно перетинались, отримаємо сферичний трикутник:

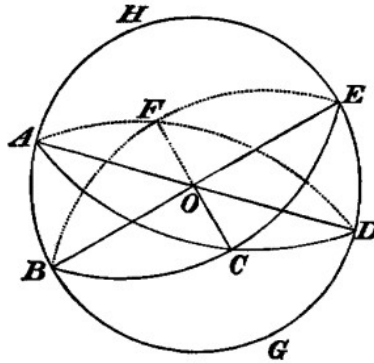


Рис. 1.3. Приклад формування сферичного трикутника

Врахувавши, що площа двокутника розраховується за формулою виду  $2Ar^2$ , можна дійти до такого висновку:  $|\triangle ABC| = (A + B + C - \pi) * r^2$ , де А, В, С – величини кутів при відповідних вершинах; r – радіус сфери. Ця формула в літературі часто носить назву "теорема Жирара" [17], опублікована французьким математиком Альбером Жираром (Albert Girard) у 1626 році.

Сферичним ексцесом (залишком) називають результат виразу  $E = A + B + C - \pi$  як величину, котра показує наскільки сума кутів при вершинах сферичного трикутника перевищує значення трансцендентного числа  $\pi$ .

Розрахувати сферичний ексцес можна деякими методами, серед яких [18]:

а) формула Каньйолі (Cagnoli)

$$\sin\left(\frac{1}{2}E\right) = \frac{\sqrt{\{ \sin s \sin(s-a) \sin(s-b) \sin(s-c) \} x}}{2 \cos \frac{a}{2} \cos \frac{b}{2} \cos \frac{c}{2}}$$

б) Теорема Льюльє (Lhuilier)

$$\tan\left(\frac{E}{4}\right) = \sqrt{\tan\left(\frac{s}{2}\right) \tan\left(\frac{s-a}{2}\right) \tan\left(\frac{s-b}{2}\right) \tan\left(\frac{s-c}{2}\right)}$$

## **1.6. Тріангуляція Делоне**

Тріангуляція Делоне для заданого набору  $P$  дискретних точок у загальному положенні є тріангуляцією  $DT(P)$ , такою, що жоден елемент  $P$  не належить внутрішній області описаного кола для кожного трикутника з  $DT(P)$ . Для можливості програмної реалізації тріангуляції Делоне використаємо алгоритм Боуера-Вотсона [19].

Алгоритм Боуера-Вотсона, по суті, є ітераційною процедурою, яка циклює по заданому набору точок і створює дійсну тріангуляцію Делоне: початковий і необхідний перший крок тут, щоб розпочати роботу, полягає у створенні великого трикутника, який міститиме в межах своїх трьох ребер усі точки, які слід тріангулювати. Створивши цей “супертрикутник” і розставивши точки, починають побудову тріангуляції. Таким чином, необхідно ітеративно додавати вершини до цієї трикутної сітки, розділяючи початковий супертрикутник на менші, доки врешті-решт не досягнемо дійсної тріангуляції Делоне. На останньому кроці потрібно буде видалити всі трикутники, які мають спільні ребра або вершини з початковим супертрикутником, оскільки вони насправді не були частиною початкового набору точок [20].

## **1.7. Формат XML та GPX для зберігання структурованої інформації**

XML (Extensible Markup Language) – мова, розроблена World Wide Web Consortium (W3C) для формалізованого визначення синтаксису для кодування документів, які можуть бути відносно швидко проаналізовані як людьми, так і машинами. Запропонований формат реалізує наданий функціонал за допомогою так званих тегів, які формують структуру документа.

Додатково, на теги також накладається функція зберігання та транспортування сформованого документу. Назви тегів обрамляються кутовими дужками (шевронами). XML використовують у різних сферах ІТ. Тут можуть зустрічатись як файли конфігурацій (налаштувань) програм, так і файли, за допомогою яких дані передаються між програмами [21].

GPX (GPS eXchange Format) — це формат для обміну даними GPS між програмами та для обміну даними GPS з іншими користувачами. На відміну від інших файлів даних, які можуть зрозуміти лише програми, які їх створили, файли GPX насправді містять опис того, що в них вбудовано, що дозволяє будь-кому створити програму, яка може читати дані всередині. GPX дозволяє обмінюватися даними з іншими користувачами GPS, які можуть використовувати ідентичне програмне забезпечення, що й ви.

Ось деякі з переваг, які надає GPX:

- GPX дозволяє обмінюватися даними з дедалі більшим списком програм для Windows, MacOS, Linux, Palm і PocketPC.
- GPX можна перетворити в інші формати файлів за допомогою простої веб-сторінки або програми-конвертера.
- GPX базується на стандарті XML, тому багато нових програм, які ви використовуєте (наприклад, Microsoft Excel), можуть читати файли GPX.
- GPX дозволяє будь-кому в Інтернеті легко розробляти нові функції, які миттєво працюватимуть із вашими улюбленими програмами [22].



## РОЗДІЛ 2. ОПИС ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕАЛІЗОВАНОГО ПРОГРАМНОГО КОМПЛЕКСУ

Процес практичної реалізації поставленої задачі виконувався за допомогою ПК Python (OpenCV та Shapely) та C#. Розроблений функціонал можна умовно поділити на опорні складові декількома способами:

- 1) використання донавченої нейромережі на вхідному зображенні,
- 2) ручне накладання знімку на карту з сервісу Google Maps по опорним точкам,
- 3) підрахунок площі.

Наведемо фінальну версію схеми, яка розкриває найголовніші деталі розроблюваного програмного комплексу:

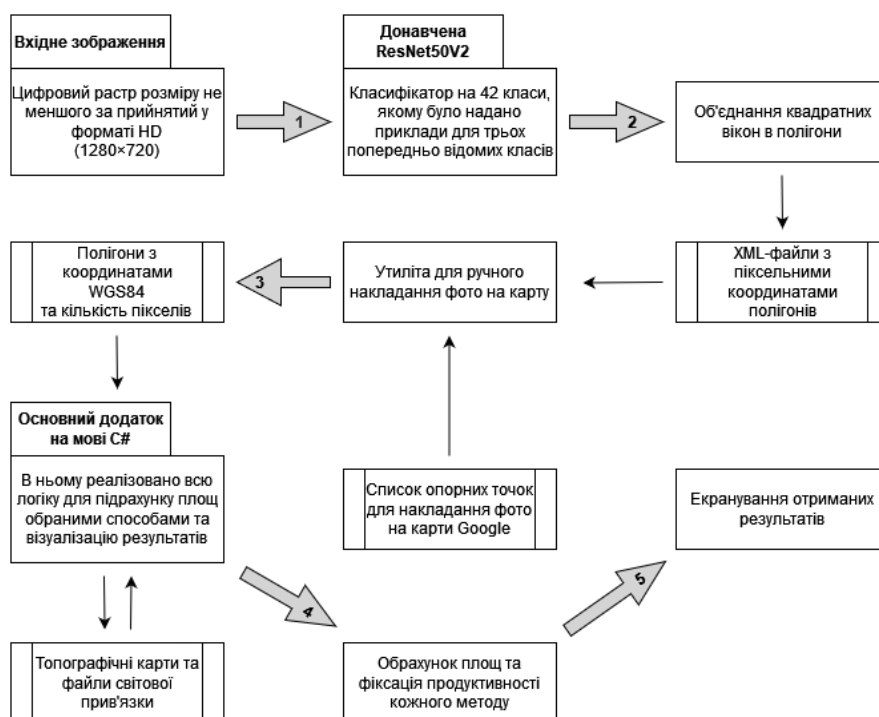
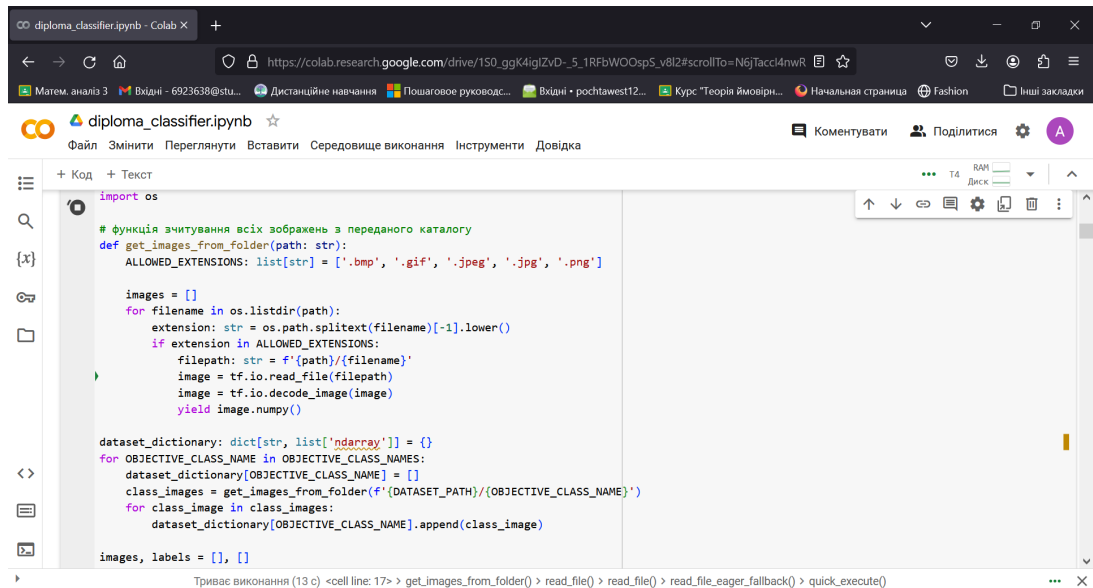


Рис. 2.1. Візуалізація послідовності дій у проєкті

Складова на C# включає неперервну взаємодію класів та їх методів, розміщених у 12-ти файлах з кодом. Тільки після розуміння необхідних до реалізації дій та проміжних етапів, можна переходити до практичної реалізації у вигляді програмного коду. Цьому й присвячено всі подальші підрозділи.

## 2.1. Навчання нейромережі

Важливим етапом навчання є підготовка коректного набору даних, після чого відбувається навчання. Хоча навчання нейромереж є досить актуальною сферою, далеко не кожен дослідник має доступ до потужних відеокарт, які значно зменшують період навчання. Тому було використано хмарний сервіс Google Colab [23], який надає тимчасові безкоштовні обчислювальні потужності відеокарти Tesla T4. Робочий сеанс в ньому виглядає так:



```
import os

# функція зчитування всіх зображень з переданого каталогу
def get_images_from_folder(path: str):
    ALLOWED_EXTENSIONS: list[str] = ['.bmp', '.gif', '.jpeg', '.jpg', '.png']

    images = []
    for filename in os.listdir(path):
        extension: str = os.path.splitext(filename)[-1].lower()
        if extension in ALLOWED_EXTENSIONS:
            filepath: str = f'{path}/{filename}'
            image = tf.io.read_file(filepath)
            image = tf.io.decode_image(image)
            yield image.numpy()

dataset_dictionary: dict[str, list['ndarray']] = {}
for OBJECTIVE_CLASS_NAME in OBJECTIVE_CLASS_NAMES:
    dataset_dictionary[OBJECTIVE_CLASS_NAME] = []
    class_images = get_images_from_folder(f'{DATASET_PATH}/{OBJECTIVE_CLASS_NAME}')
    for class_image in class_images:
        dataset_dictionary[OBJECTIVE_CLASS_NAME].append(class_image)

images, labels = [], []
```

Рис. 2.2. Робота у сервісі Google Colab

В ході експериментів набуто наступні результати донавчання впродовж 30 епох у вигляді графіків, де синій колір — тренування, жовтий — валідаційний:

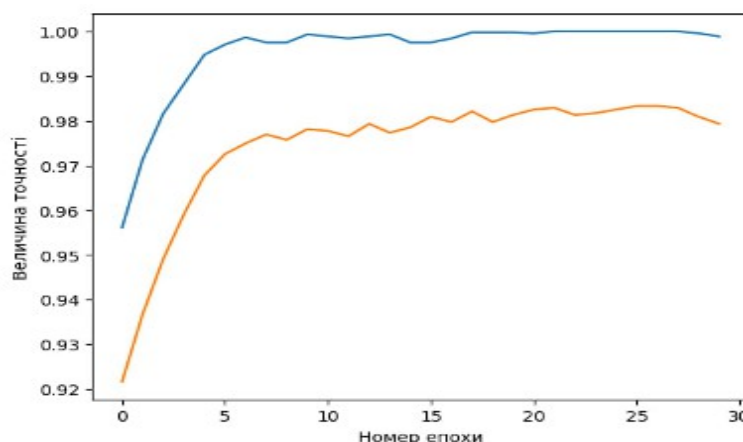


Рис. 2.3. Графіки точностей навчання та тестування

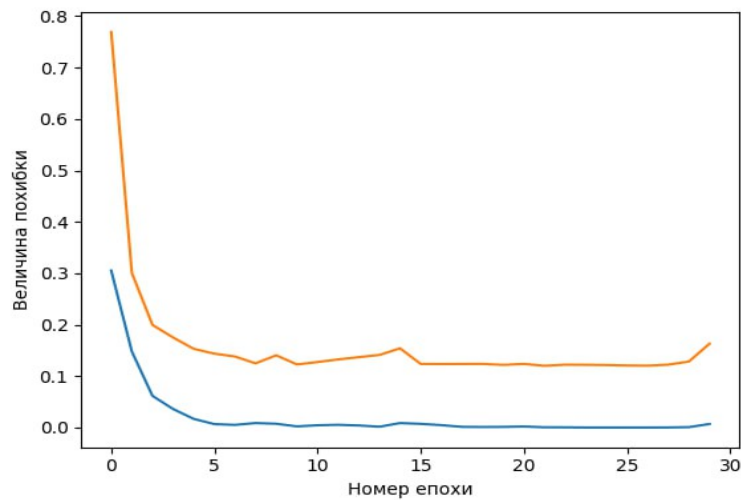


Рис. 2.4. Графіки величин похибок, отримані під час навчання та валідації

Далі, на основі принципу “ковзного вікна” було розглянуто два способи пошуку полігонів, що оточують руйнування:

1) Побудова сітки. Будувалась сітка з прямокутників прийнятної для обраної неймережі розміру (64x64). Кожен такий прямокутник отримував відповідь щодо приналежності до певного класу, після чого відбирались лише ті з них, котрі були розпізнані як зруйновані будинки. Прямокутники одного класу, розташовані поруч, об'єднувались в суцільний полігон.



Рис. 2.5. Приклад №2 без зсуву

2) Наведена в попередньому пункті оцінка площі не завжди є прийнятною, з причини недостатньої дискретизації. Для вирішення можна діяти так:

- обравши довільним чином декілька значень для зсувів, зміщувати умовний початок координат на обрані величини для кожної з вісей так, щоб кожен зсув був приміненний для кожної з них,
- фрагменти зображення класифіковані як зруйновані будинки ставали складовими суми добутків квадрату сторони ковзного вікна та поточної ймовірності до віднесення цього ж класу,
- оптимальні зсуви визначались шляхом вибору тих значень, котрі дають площу найближчу до середньої з них.

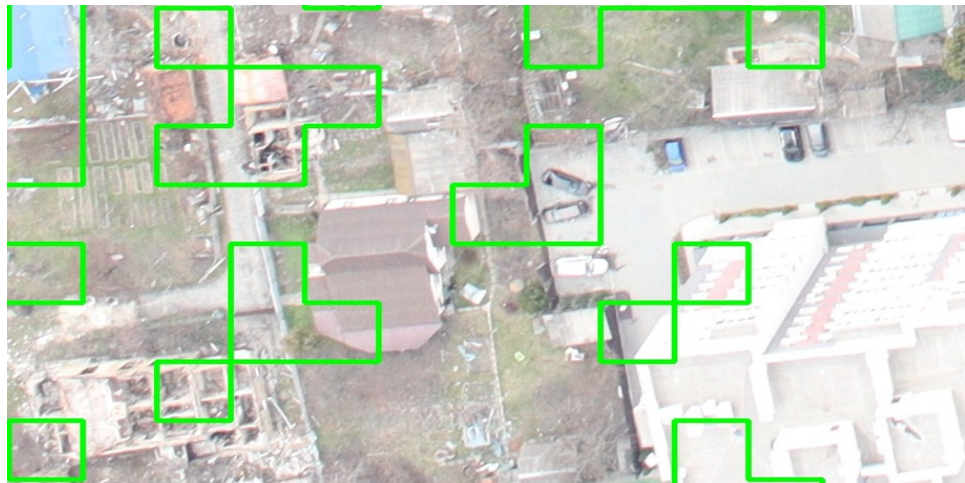


Рис. 2.6. Приклад №2 із зсувом

Пояснити доцільність використання формули Герона для оцінки площ трикутників на поверхні еліпсоїда в нашому випадку можна, навівши таблицю:

Таблиця 3

Розбіжність між площами на площині та на сфероїді

| Перший катет, км | Другий катет, км | Площа на еліпсоїді, км <sup>2</sup> | Площа на площині, км <sup>2</sup> | Розбіжність, % |
|------------------|------------------|-------------------------------------|-----------------------------------|----------------|
| 2                | 3                | 3                                   | 3                                 | 0              |
| 20               | 30               | 300.001                             | 300                               | 0.01           |
| 200              | 300              | 30007.986                           | 30000                             | 0.03           |
| 500              | 750              | 187812.452                          | 187500                            | 0.17           |
| 1000             | 1500             | 755024.350                          | 750000                            | 0.67           |
| 2000             | 3000             | 3081254.79                          | 3000000                           | 2.64           |

Всі розрахунки в таблиці вище проводились на широті, рівній значенню десяткового градуса в 50, що майже співпадає із широтою київської області.

Отже, для запропонованих вхідних параметрів, різниця між площею трикутника на площині та на еліпсоїді стає помітною лише при перевищенні площі фігури значення, рівного 300 км<sup>2</sup>, що робить можливим вважати використовуваний в програмі підхід коректним, оскільки тестування проводились на значно менших розмірах цільових областей.

## 2.2. Реалізація файлів світової прив'язки

Частина проєкту реалізована на С# відповідає за дві основоположні складові утвореного програмного рішення — маніпуляція над world-файлами та підрахунок площ попередньо передбачених полігонів із потрібним вмістом. В даному підрозділі зупинимось на розгляді першого з них.

### 2.2.1. Ілюстрований опис ручного створення

Одразу після запуску програми на екрані користувача відобразиться головне вікно. Для відкриття растрової карти в додатку необхідно виконати таку ланцюжкову послідовність у меню: “Файл” → “Відкрити...” → “Карту” або скористатись комбінацією клавіш “Ctrl + M”:

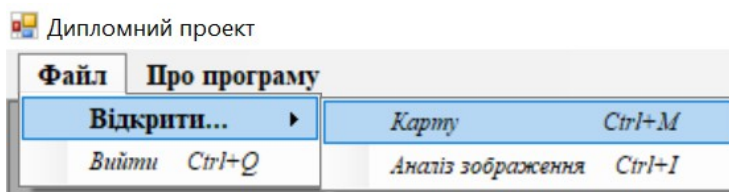


Рис. 2.7. Складові головного меню розробленої програми

Далі відобразиться модальне діалогове вікно для вибору карти (для зручності, пошук розпочинається із каталогу з виконуваною програмою). У випадку підтвердження вибору існуючого файлу, котрий може бути зчитаний програмою, вміст форми успішно зміниться (рис. 2.8):



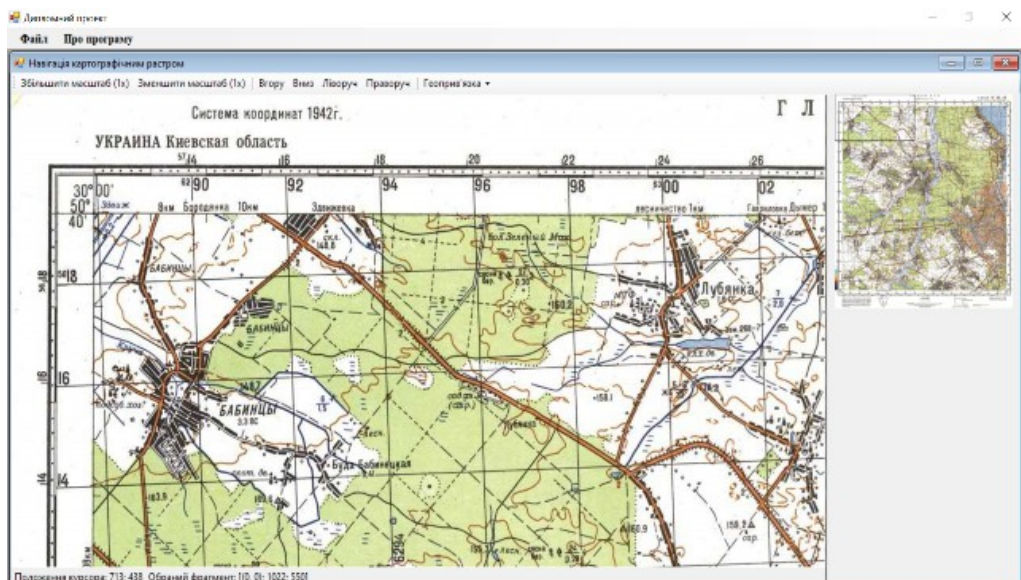


Рис. 2.8. Отриманий результат після вибору карти для опрацювання

На рисунку видно відкрите вікно із обраною картою, котра займає більше половини новоутвореної робочої області. Також присутня та ж карта, проте у зменшеному розмірі з правої сторони. Також є меню у верхній частині, кнопки якої відносно повністю розкривають власне призначення.

### 2.2.2. Повний приклад програмного нанесення треку

Обравши “Геоприв'язка” → “Допоміжна інформація”, відкриватиметься вікно для вибору трьох опорних точок файлу світової прив'язки (див. рис. 2.9).

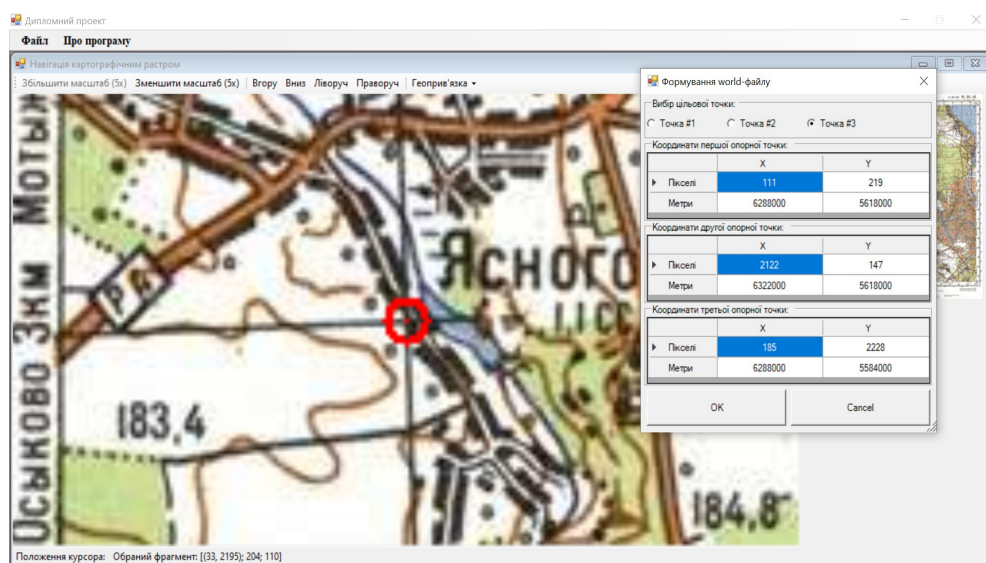


Рис. 2.9. Вигляд під час ручного формування world-файлу

Загалом, в ході виконання роботи аналогічним чином було оформлено геоприв'язку для 19 карт різних масштабів та місцевостей на них (здебільшого, Київська та Житомирська області) у таких співвідношеннях:

- а) 11 карт масштабу 1:100000 (в одному сантиметрі один кілометр);
- б) 6 карт масштабу 1:200000 (в одному сантиметрі два кілометри);
- в) 2 карти масштабу 1:500000 (в одному сантиметрі п'ять кілометрів).

### 2.2.3. Нанесення треків з побутових пристроїв на цифрові карти

Процедура вибору треку для накладання мало чим відрізняється від вибору карти для аналізу. Єдина відмінність у файлі, який програмний комплекс може сприймати в даний момент — замість графічних файлів шукаються файли з GPX-розширенням.

Отже, після виконання всіх проміжних розрахунків, робоча область додатку виглядатиме приблизно так:

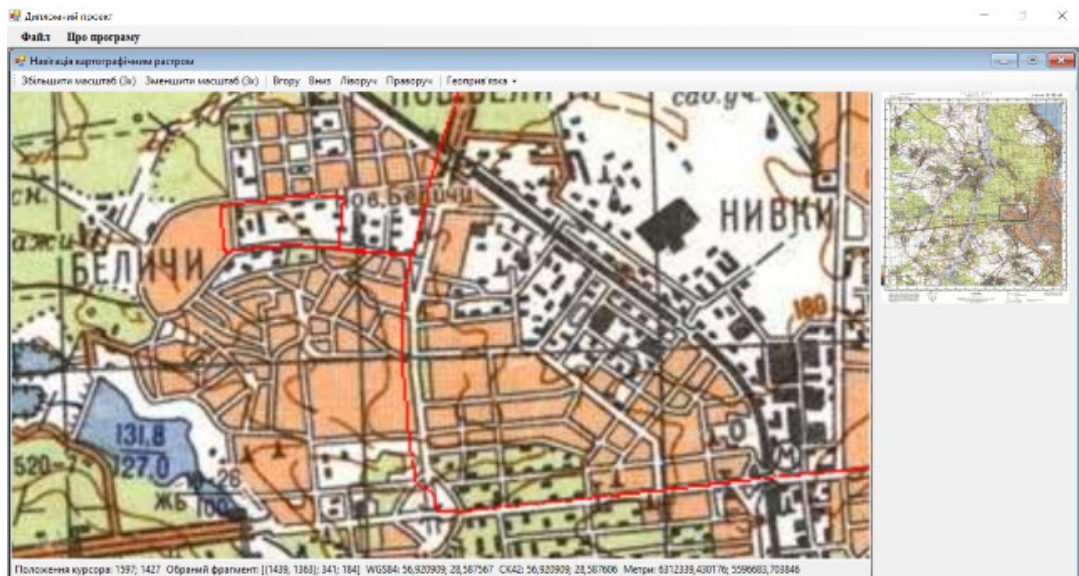


Рис. 2.10. Вигляд графічного полотна з обраним треком

Після накладання точок треку користувач може оцінювати положення довільного пікселя в трьох цільових системах координат (WGS84, СК42, плоскі прямокутні координати Красовського) шляхом звичайного перенесення курсору миші по графічній площині, що стало можливим завдяки формулам 1.1 - 1.16.

### 2.3. Опис методів підрахунку площ

Ілюстративні діаграми (блок-схеми) можна було б сформувати й для інших алгоритмів проекту. Проте, враховуючи що ідея використання псевдокоду для коментування ПЗ неодноразово згадується в різноманітних роботах, присвячених розробці ПЗ. Більш зручним для використання в якості псевдокоду представляються створені для формальних методів розробки ПЗ мови формальних специфікацій. Вони частіше за все мають готові системи ASCII - позначень для багатьох математичних визначень. Таким чином, їх можна використовувати у якості псевдокоду в коментарях до програм, в системах генерації документації до ПЗ. Крім того, мови формальних специфікацій зазвичай мають засоби (spellchecker) для перевірки синтаксису й виконання статичної типизації. Зокрема, виконувалось документування функції (метода класу) підрахунку площі трикутника на еліпсоїді [24], де функції підрахунку площі не використовували цикли, для наближення специфікації до коду аналогом оператора - виразу було використано вираз let. Відповідна схема мови RSL читається достатньо прозоро:

```
area1 (al, be, ga, R) is
  let s = (al + be + ga) / 2.0,
      root = sqrt(tan (s / 2.0) * tan((s- al)/2.0) *
                  tan((s - be) /2.0) * tan((s - ga) /2.0)),
      eps = 4.0 * arctan (root)
  in eps * R * R end
end
```

Вона мало чим відрізняється від коду відповідного методу на C#:

```
double area1 ( double al, double be, double ga, double R)
{
    double s = (al+be+ga)/2.0;
    double root = Math.Sqrt (Math.Tan(s/2.0) * Math.Tan((s-al)/2.0) *
                              Math.Tan((s-be)/2.0) * Math.Tan((s-be)/2.0));
    double eps = 4.0 * Math.Atan(root);
    return eps * R * R;
}
```



В результуючій програмі у явному вигляді було залучено наступні методи підрахунку площ:

- 1) На основі пікселів (множення кількості наявних пікселів у поточному полігоні на оцінену площу одного пікселя на зображенні).
- 2) Спеціальний геодезичний алгоритм [25], котрий дозволяє зробити можливим обрахунок площі сферичного багатокутника над одиничною сферою.
- 3) Із залученням тріангуляції Делоне на площині (запустити процес тріангуляції на точках у метрах та шукану площу оцінювати як суму площ одержаних трикутників за формулою Герона).
- 4) Застосувавши тріангуляцію Делоне на сфері (те саме, що й у попередньому, тільки в ньому площа кожного трикутника рахується як сферична).

В програмі для підрахунку площ оцінених багатокутників різної форми можна зробити вибір папки для опрацювання:

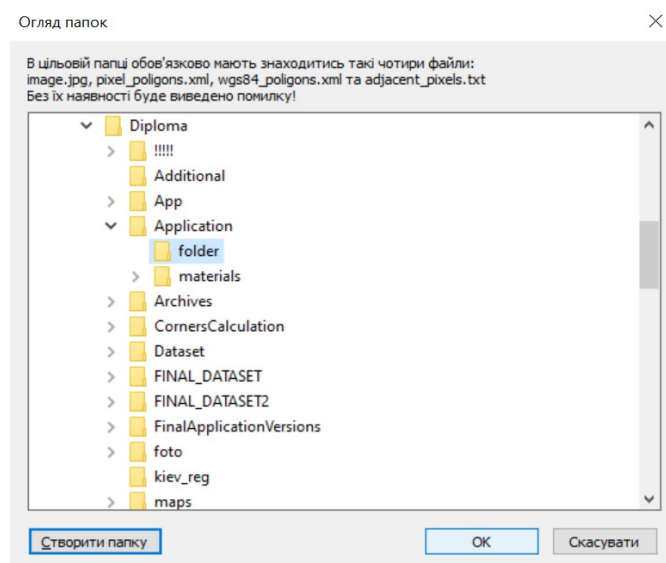


Рис. 2.11. Процес вибору папки для опрацювання

Після вибору каталогу із всіма необхідними файлами програма поверне результат необхідних розрахунків у передбаченому для цього форматі.

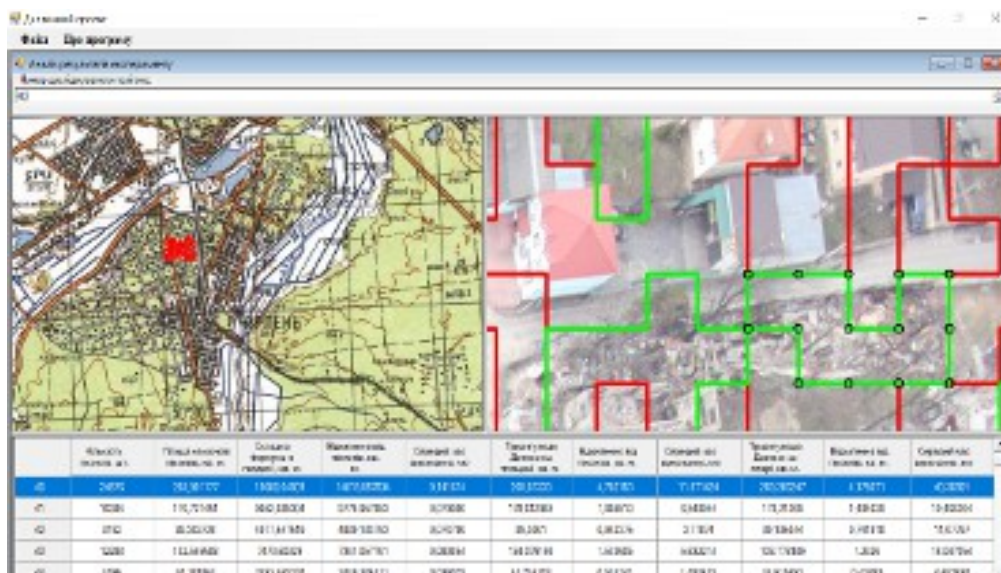


Рис. 2.12. Результат обрахунків (приклад 1)

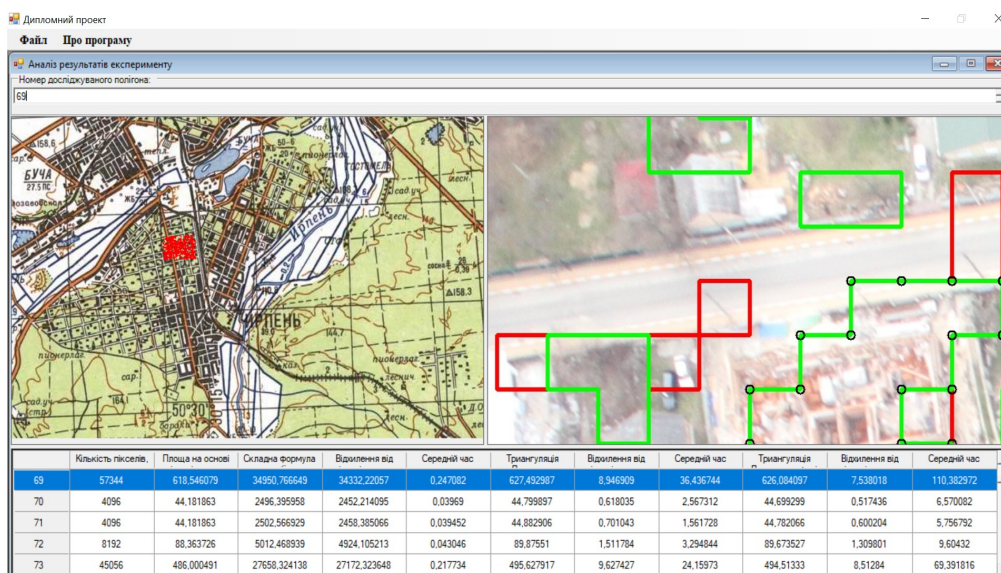


Рис. 2.13. Результат обрахунків (приклад 2)

В робочій області програми, можна обирати номер полігону для аналізу, що передбачає зміну обрамленого полігону у правій частині форми й наведення фокусу керуючого елемента з таблицею на той самий по номеру рядок.

## ВИСНОВКИ

Впродовж виконання дослідження по темі роботи було:

1. Обраховано площу зруйнованих будинків за допомогою нейромережевого класифікатору.
2. Для покращення результатів класифікації, після анотування додаткових прикладів, донавчено преднавчену нейромережеву модель ResNet50V2 шляхом додавання допоміжних прикладів.
3. Застосовано принцип ковзного вікна при аналізі вхідного високоякісного зображення.
4. Забезпечено обмін та збереження згенерованих даних в XML-форматі.
5. Після підготовки чотирьох опорних точок вирішено обернену геодезичну задачу при пошуку розміру пікселя в метрах.
6. Власноруч запрограмовано методи двостороннього переходу як для систем WGS84 та СК42, так і для координат СК42 й плоских прямокутних координат (метрів).
6. Використано три методи обрахунку площ полігону, координати вершин якого задані в системі WGS84.
7. Порівняно швидкість обрахунку площ, які здатні надати обрані методи.
8. Реалізовано додаток з графічним інтерфейсом.

Перспектива подальших досліджень може полягати у:

- перевірці роботи проєкту на іншій місцевості (регіоні, країні тощо) після внесення правок до декотрих географічних складових (системи координат, референц-еліпсоїди, формули переходу тощо);
- безпосередньо практичному застосуванні на державних та комерційних підприємствах для фінансової оцінки збитків від пошкодження посівних полів, житлових масивів, автомагістралей та інших об'єктів інфраструктури, з залученням необхідних впроваджень, для їх усунення та майбутньої реконструкції.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Оцінка [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Оцінка>.
2. Основи екології: підручник — Олійник Я. Б., Шищенко П.Г. К. : Знання, 2012., 558 с
3. Геодезія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Геодезія>.
4. Штучна нейронна мережа [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Штучна\\_нейронна\\_мережа](https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа).
5. Простий путівник, який допоможе зрозуміти ІІІ [Електронний ресурс] - Режим доступу до ресурсу: <https://www.bbc.com/ukrainian/resources/idt-74697280-e684-43c5-a782-29e9d11fecf3>.
6. The Turing Test [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.investopedia.com/terms/t/turing-test.asp>.
7. Machine Learning, ML [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>.
8. Training Loss, Validation Loss and Evaluation Loss [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@penpencil.blr/what-is-the-difference-between-training-loss-validation-loss-and-evaluation-loss-c169ddeccd59>.
9. Backpropagation [Електронний ресурс] – Режим доступу до ресурсу: <https://brilliant.org/wiki/backpropagation/>.
10. What is gradient descent? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/topics/gradient-descent>.
11. ML | Stochastic Gradient Descent (SGD) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>.
12. Sliding Window Technique [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/sliding-window-technique/>.

13. Основні співвідношення сфероїдальної геодезії [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5646478/>.
14. Вища геодезія. Підручник/ Савчук С.Г. – Житомир, 2005. – 315 с.
15. Іванько Т. В. Інерціально-супутникова навігаційна система безпілотного літального апарата [Електронний ресурс] / Тамара Валеріївна Іванько. – 2020. – Режим доступу до ресурсу: <https://ela.kpi.ua/server/api/core/bitstreams/884b877a-08b8-45f4-852b-59e845a0c653/content>.
16. World files for raster datasets [Електронний ресурс] – Режим доступу до ресурсу: <https://pro.arcgis.com/en/pro-app/latest/help/data/imagery/world-files-for-raster-datasets.htm>.
17. Girard's Theorem [Електронний ресурс] – Режим доступу до ресурсу: [www.theoremoftheday.org/GeometryAndTrigonometry/Girard/TotDGirard.pdf](http://www.theoremoftheday.org/GeometryAndTrigonometry/Girard/TotDGirard.pdf).
18. Spherical Trigonometry For the Use of Colleges and Schools [Електронний ресурс]. - Режим доступу до ресурсу: <https://www.gutenberg.org/ebooks/19770>.
19. Bowyer–Watson algorithm [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Bowyer–Watson\\_algorithm](https://en.wikipedia.org/wiki/Bowyer–Watson_algorithm).
20. Bowyer-Watson Algorithm for Delaunay Triangulation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.gorillasun.de/blog/bowyer-watson-algorithm-for-delaunay-triangulation/>.
21. Формат даних XML [Електронний ресурс] – Режим доступу до ресурсу: <https://javarush.com/ua/quests/lectures/ua.jru.module2.lecture28>.
22. GPX Explained [Електронний ресурс] – Режим доступу до ресурсу: [https://www.topografix.com/gpx\\_for\\_users.asp](https://www.topografix.com/gpx_for_users.asp)
23. Google Colaboratory [Електронний ресурс] – Режим доступу до ресурсу: <https://colab.google/>.
24. ...
- ...
- ...
2024. ...

25. lcх366 / SphericalPolygon [Электронный ресурс] – Режим доступа до ресурсу:  
[https://github.com/lcx366/SphericalPolygon/blob/master/sphericalpolygon/excess\\_area.py](https://github.com/lcx366/SphericalPolygon/blob/master/sphericalpolygon/excess_area.py).