МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНИ СЕМЕНА КУЗНЕЦЯ

ЗВІТ
о виконанні лабораторної роботи №3
з дисципліни «Захист інформації»
за темою «Невидимі дані у власних файлах»

Виконав:
Студент 4 курсу
групи 6.04.122.010.22.1
факультету ІТ
Холоша Сергій

Перевірив:
Професор кафедри
кібербезпеки та ІТ
Тютюнник В. В.

Харків – 2025

Порядок виконання практичної частини

Крок 1. Дослідження готових стеганографічних інструментів

1) Завантажити програму Steghide або скористатися онлайн-сервісами стеганографії. Інструкція з встановлення та використання Steghide https://labex.io/tutorials/hide-data-in-steghide-549941;

2) Спробувати приховати текстове повідомлення в зображенні;

3) Витягнути приховане повідомлення та переконатися в його цілісності;

4) Опанувати основні принципи роботи LSB-стеганографії.

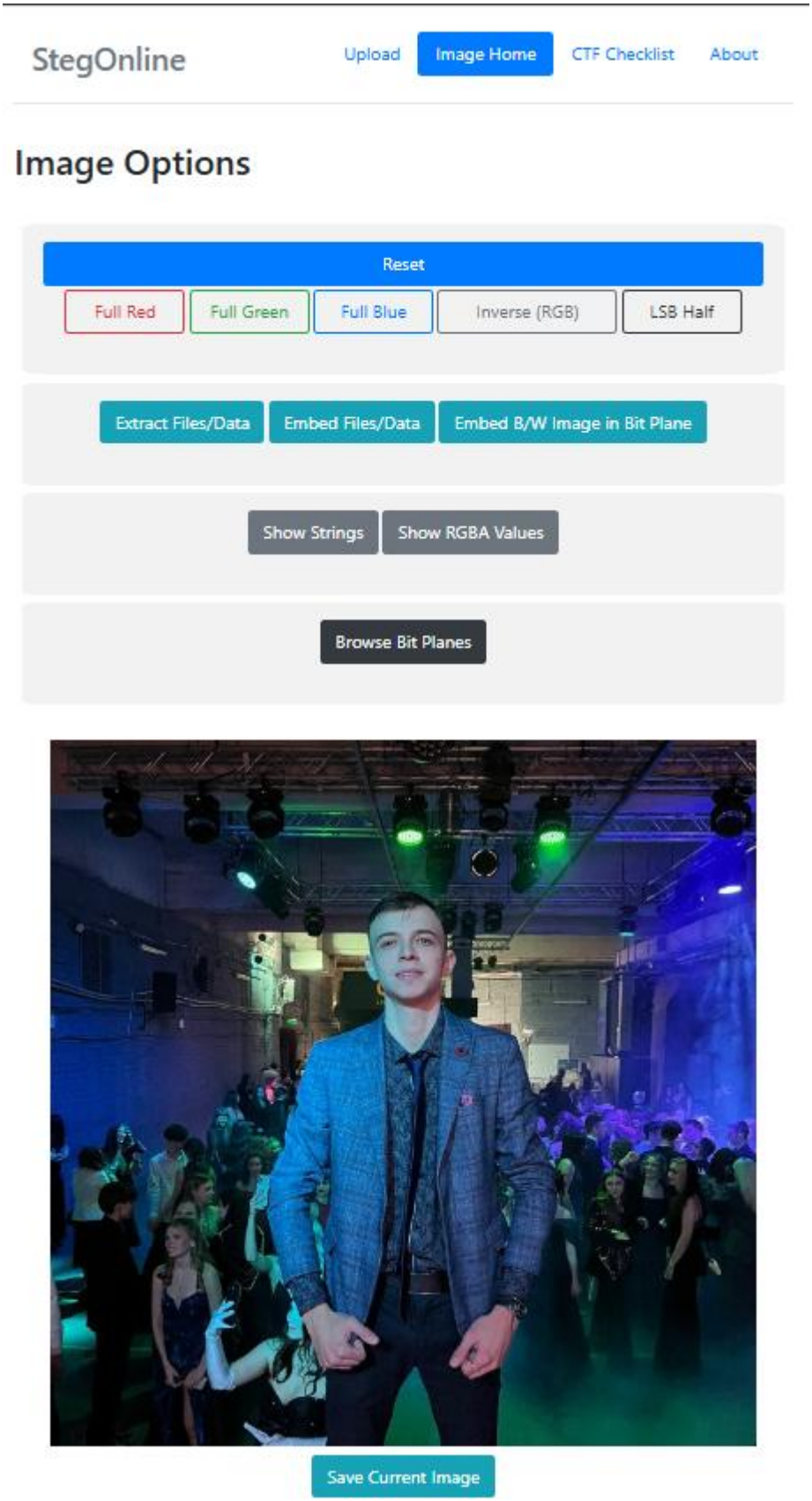

Рис. 1. Зображення завантажене на сервіс для стеганографії

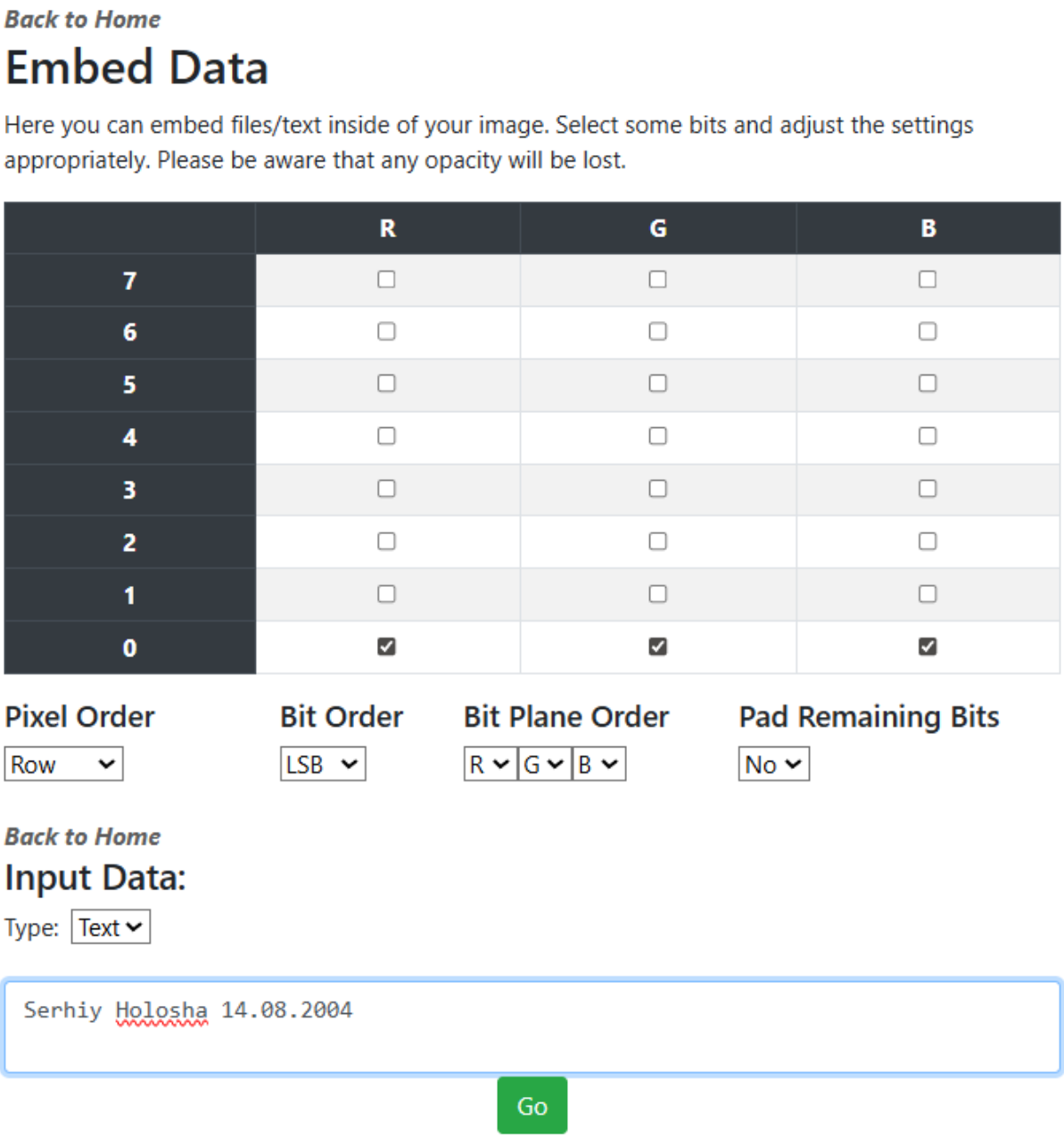

Рис. 2. Запис даних за методом LSB

Output

Use the "Save Image" Button to download the image, as saving through your browser's right-click menu may result in inconsistent data.

Download Extracted Data

Рис. 3 Отримане зображення

Extract Data

Here you can extract data hidden inside of the image. Select some bits and adjust the settings appropriately. The final extracted data is checked against some basic file headers, and so the filetype can be automatically determined.
Please note that Alpha options are only available if the image contains transparency.

	R	G	B
7	☐	☐	☐
6	☐	☐	☐
5	☐	☐	☐
4	☐	☐	☐
3	☐	☐	☐
2	☐	☐	☐
1	☐	☐	☐
0	☑	☑	☑

Pixel Order: Row

Bit Order: LSB

Bit Plane Order: R G B

Trim Trailing Bits: No

Go

Results

No file types identified.

The results below only show the first 2500 bytes. Select "Download" to obtain the full data.

Ascii (readable only):

```
Serhiy H  olosha 1  4.08.200  4.q....c  .$.M..q.  m..v....  ....6u..
@.......  +.....cz  ....p...  j.$.rH.m  ..m....O  .q.J....  .[.A}...
..E...vI  !...AF..  P.*..+.D  p=.[...q  .....I..  ........  .H\8.$..
```

Hex (Accurate):

```
536572686979204 86f6c6f7368612031342e30382e32303034ec71d8e3b6db638924ec4da4
9271c76d8bbc762ef20e8e1b1388a9367580f1401981dcbb83adbe2bb5ceede7a7637aff8f
c0fc7004f6d56ab524957248db6db71b6da8df189e4fe0711a4aa38e1fca855b1c417dc8db
```

Рис. 4 Отримані назад дані

Крок 2. Приховування даних у зображенні

1) Обрати власне фото або улюблене зображення;

2) Приховати повне ім’я та дату народження у файлі;

3) Використати онлайн-інструменти або готові програми для стеганографії;

4) Порівняти оригінал та модифіковане зображення візуально;

5) Провести детальний порівняльний аналіз створеного стегоконтейнера:

– Перейти на веб-сайт https://www.img2go.com/compare-image;

– Завантажити оригінальне зображення та власний стегоконтейнер;

– Проаналізувати карту відмінностей з підсвіткою;

– Вивчити числові метрики: процент відмінностей, кількість змінених пікселів по каналах RGB;

– Зробити висновки про візуальну непомітність внесених змін та ефективність стеганографії.

Рис. 5 Зображення до стеганографії

Рис. 6 Зображення після стеганографії

Similarity
99.98%

Method
Structural Similarity Index Measure

	First file	Second file
Name	ciphered.png	pic.jpg
Size	875.89 KB	118.06 KB
Height	640px	640px
Width	640px	640px
File type	png	jpeg

Рис. 7 Порівняння за допомогою рекомендованого веб-сайту

Аналіз карти відмінностей показує надзвичайно високу ефективність застосованого стеганографічного методу. Структурний індекс подібності (SSIM) становить 99.98%, що свідчить про майже повну ідентичність оригінального та модифікованого зображень. На карті відмінностей видно лише окремі розсіяні червоні пікселі на білому фоні, що вказує на мінімальні локальні зміни в структурі зображення. Враховуючи розмір зображення

640×640 пікселів (409,600 пікселів загалом), візуально змінені пікселі становлять менше 0.5% від загальної кількості, що робить їх абсолютно непомітними для людського ока.

Різниця в розмірах файлів (875.89 KB для PNG з вбудованими даними проти 118.06 KB оригінального JPEG) пояснюється зміною формату та методу стиснення, а не обсягом прихованої інформації. Відсутність будь-яких візуально помітних артефактів, шуму чи спотворень на модифікованому зображенні підтверджує високу якість реалізації LSB-методу. Такі результати демонструють успішне досягнення основної мети стеганографії - забезпечення непомітності вбудованої інформації при збереженні візуальної цілісності контейнера.

З точки зору практичної ефективності, отримані метрики (SSIM > 99.9%, візуальні відмінності < 0.5%) вказують на те, що стеганосистема успішно пройшла б базові тести на візуальну непомітність і могла б протистояти простим методам стегоаналізу, заснованим на візуальному порівнянні або статистичному аналізі гістограм першого порядку.

Крок 3. Пошук прихованих повідомлень

1) Завантажити 2–3 зображення з відкритих джерел (Unsplash, Pixabay);
2) Спробувати виявити приховану інформацію за допомогою онлайн-інструментів;
3) Використати різні методи пошуку: аналіз LSB, гістограм, метаданих;
4) Задокументувати процес та використані інструменти.

Рис. 8 Перше зображення для пошуку прихованого повідомлення

Рис. 9 Друге зображення для пошуку прихованого повідомлення

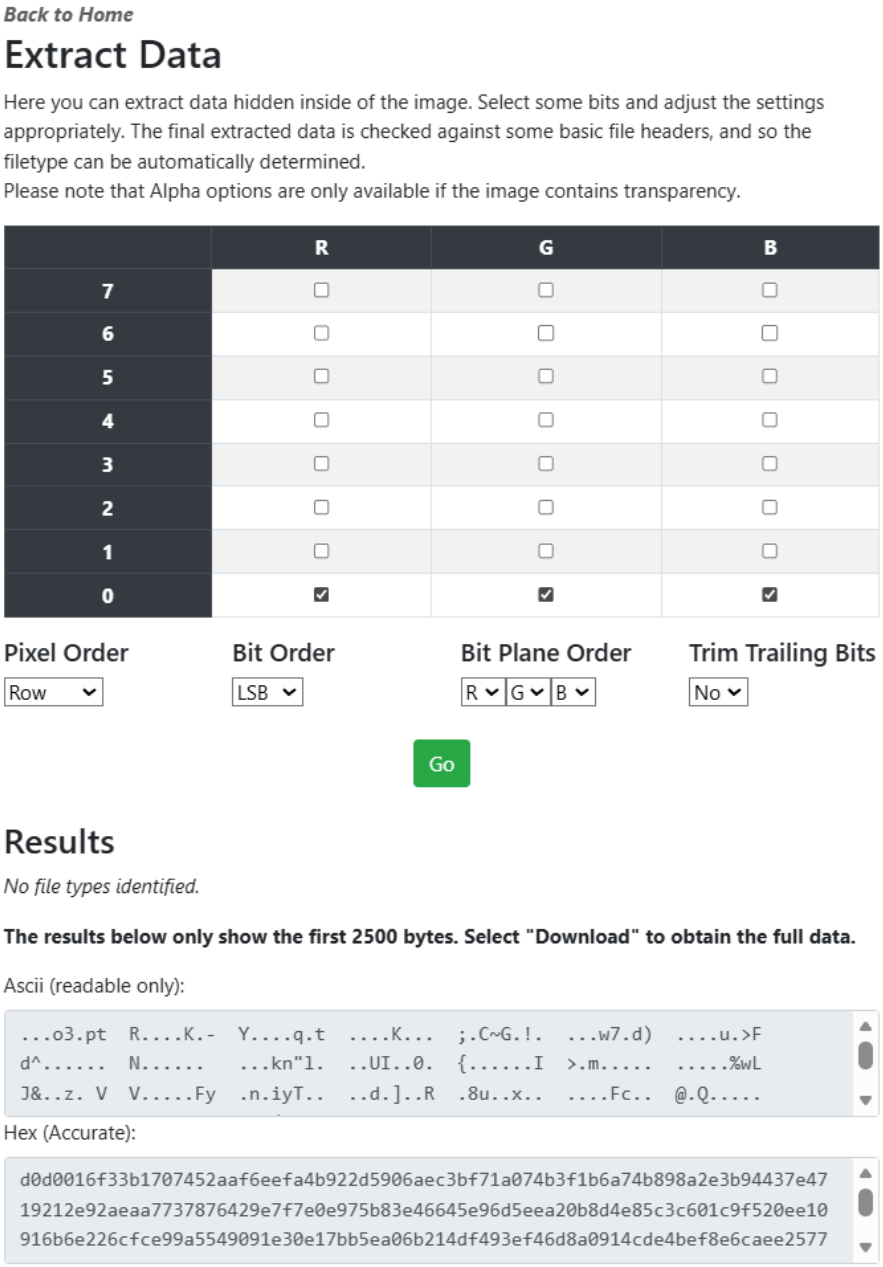

Рис. 10 Спроба отримати дані через стеганографію з першого зображення

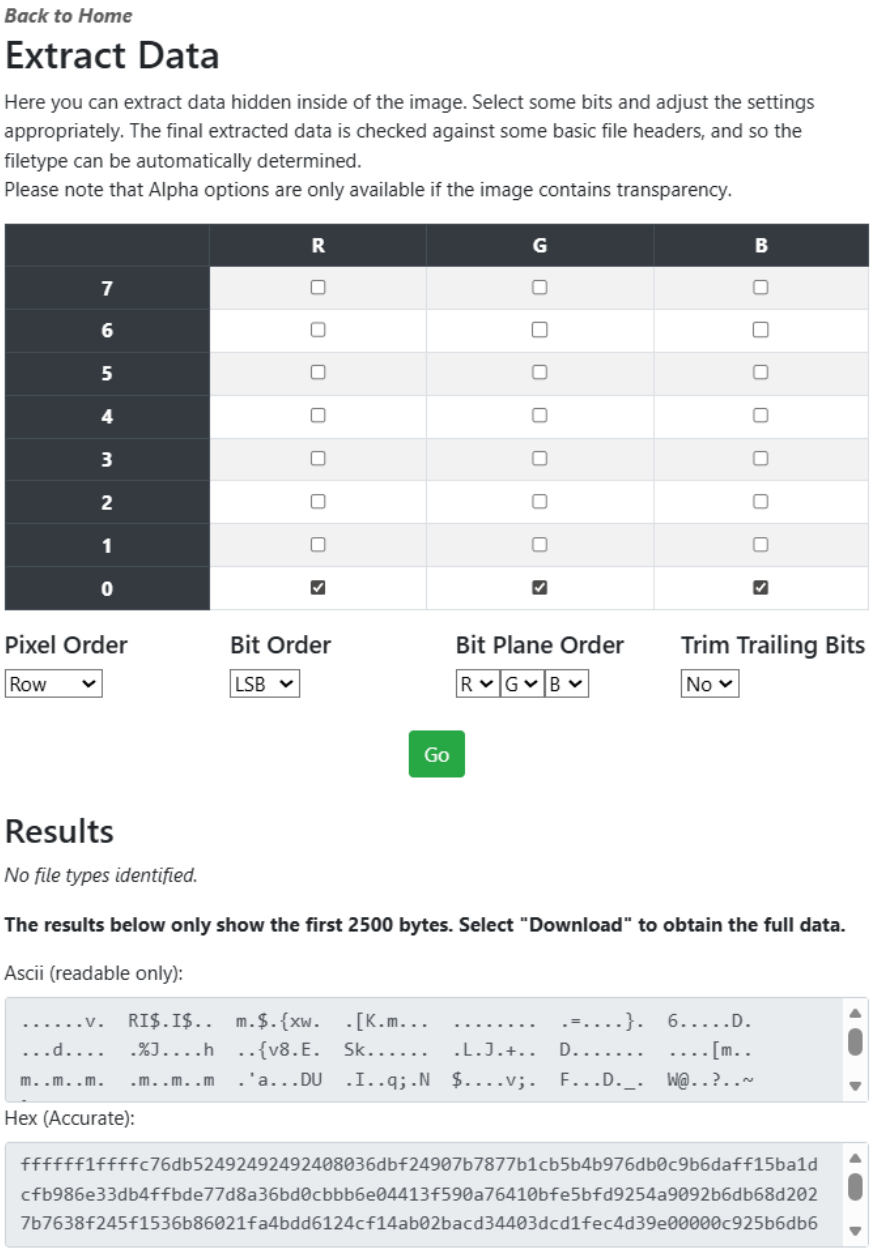

Рис. 11 Спроба отримати дані через стеганографію з другого зображення

Для аналізу гістограми використаємо безкоштовний аналог PhotoShop – Canva Affinity.

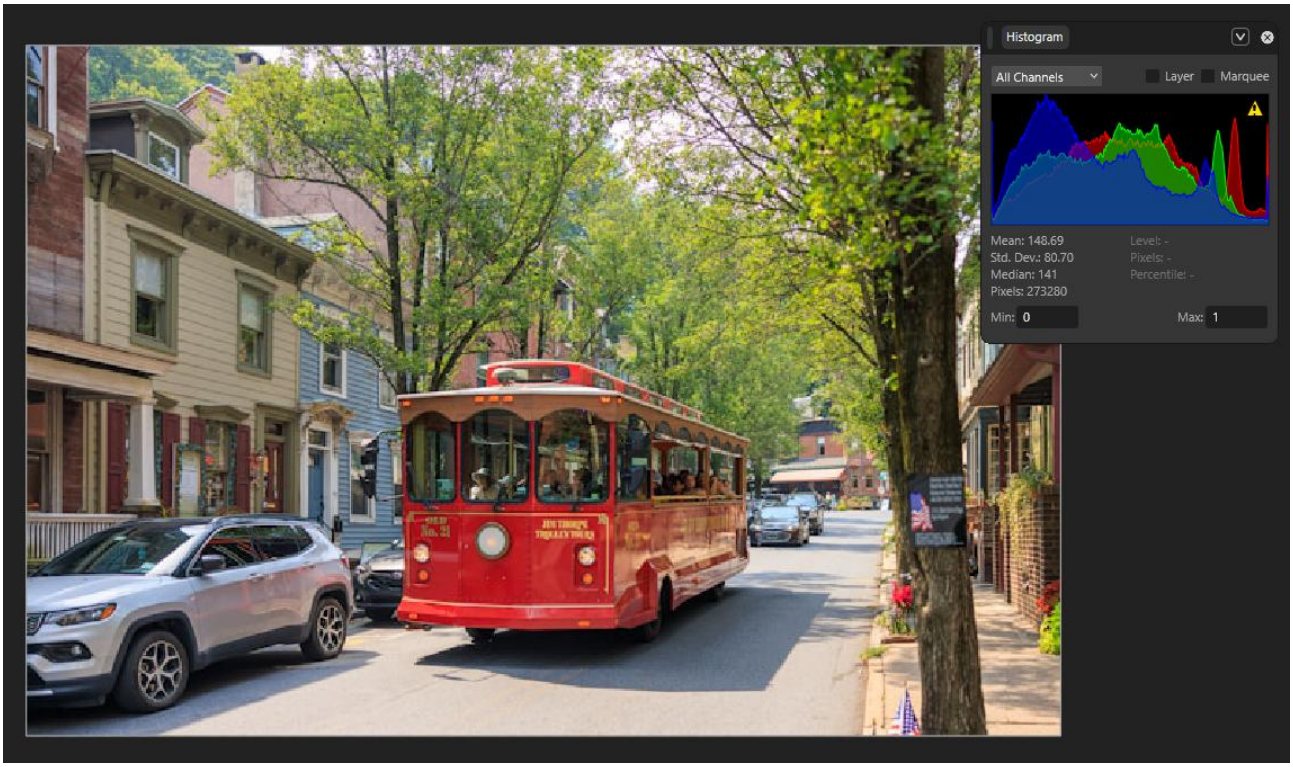

Рис. 12 Перше зображення

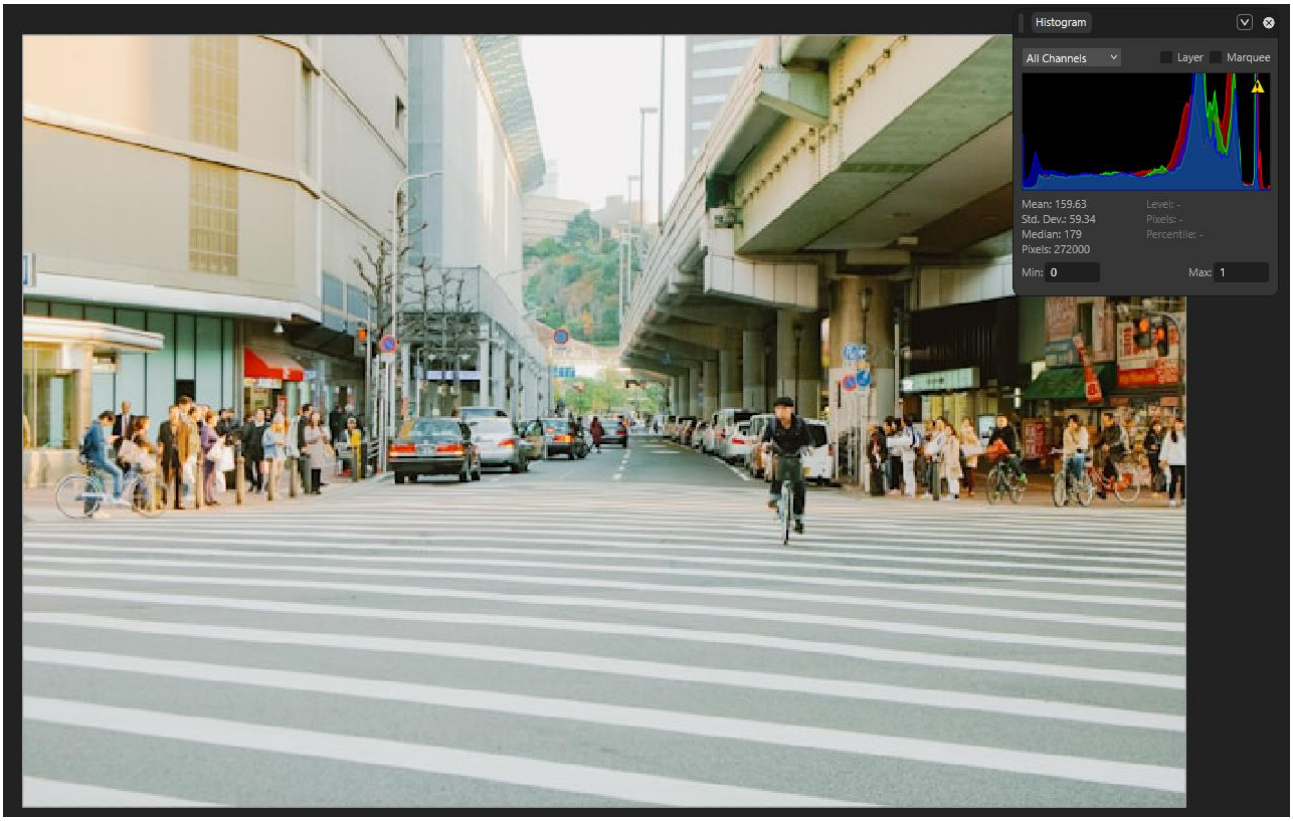

Рис. 13 Друге зображення

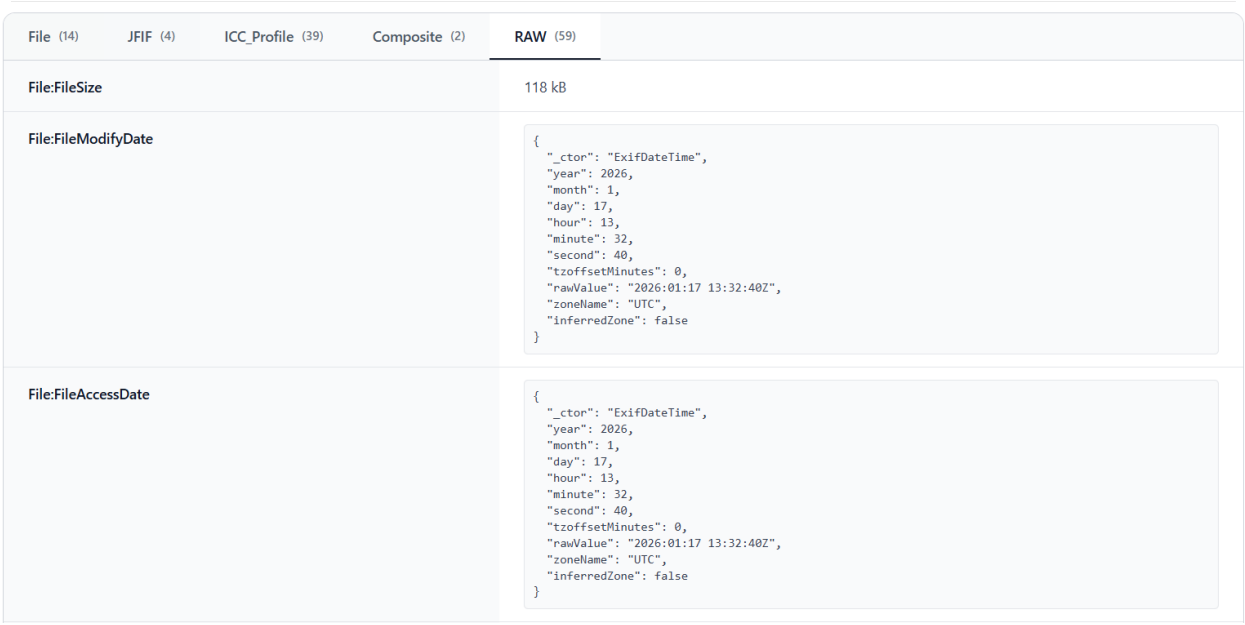

Рис. 14 Аналіз метаданих першого зображення

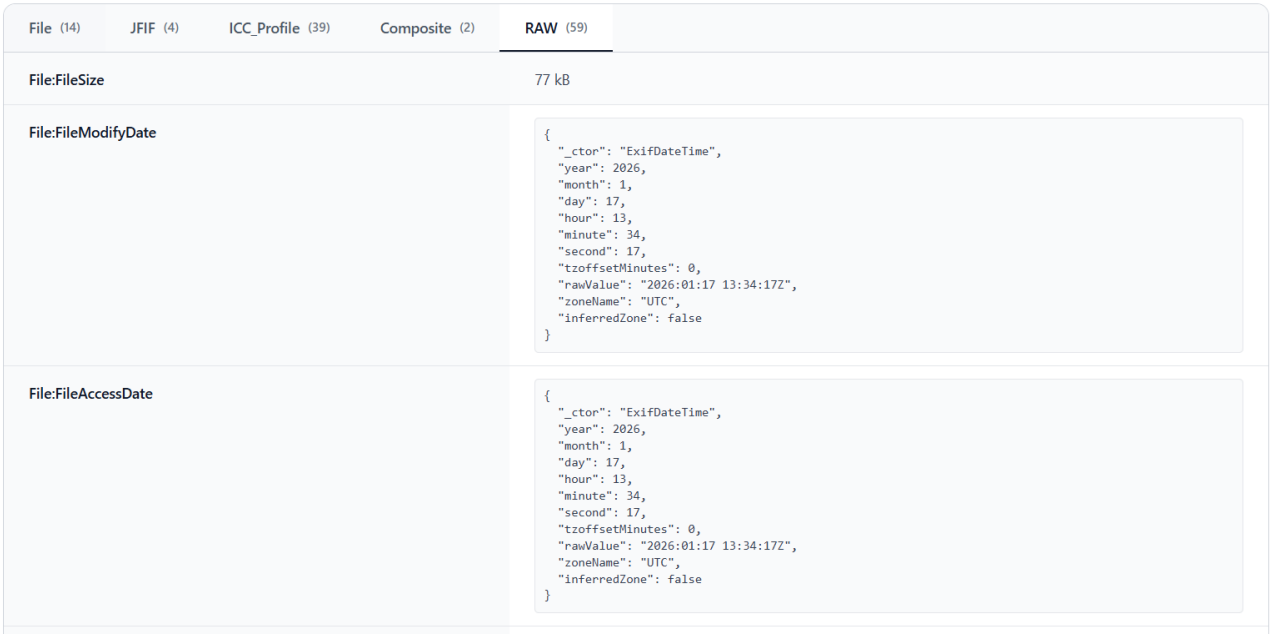

Рис. 15 Аналіз метаданих другого зображення

Отже, в результаті аналізу LSB, гістограм та метаданих не було виявлено присутності використання стеганографії.

Крок 4. Аналіз метаданих

1) Дослідити EXIF-дані власних фотографій;
2) Визначити потенційні витоки особистої інформації через метадані;
3) Очистити метадані перед публікацією в соціальних мережах.

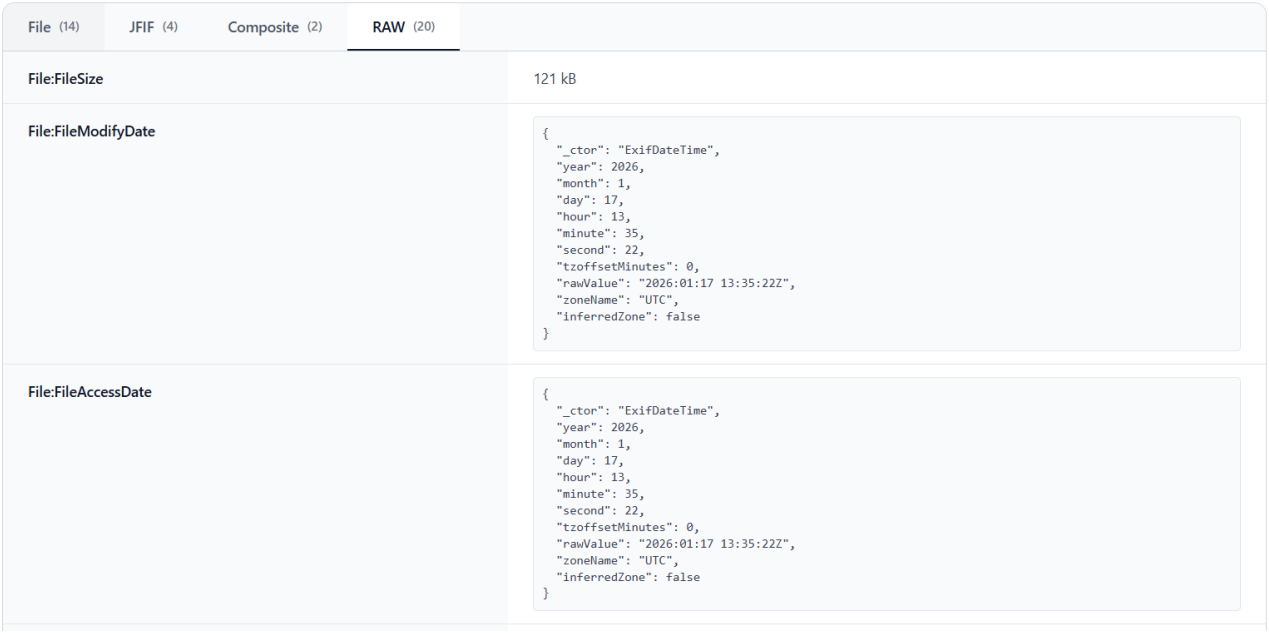

Рис. 16. EXIF дані мого зображення

Дані не містять чутливої інформації

Технічне завдання

Завдання: Реалізувати програму для приховування текстової інформації в зображеннях з демонстрацією розуміння принципів стеганографії.

Обов'язкові функціональні вимоги:

– Етап 1: Продемонструвати розуміння процесу через покроковий алгоритм;

– Етап 2: Реалізувати функції hide_message() та extract_message();

– Етап 3: Показати роботу на власних персональних даних;

– Етап 4: Провести аналіз змін в зображенні (розмір, візуальні відмінності).

Технічні вимоги:

– Можна використовувати готові бібліотеки для роботи з файлами зображень (PIL, Canvas, System.Drawing);

– **НЕ можна** використовувати готові бібліотеки стеганографії (stegano, steghide тощо);

– Потрібно самостійно реалізувати логіку приховування даних.

```
Оригінальне повідомлення: My name is Serhii Holosha and I love programming!
Довжина: 49 символів
Бінарне представлення (перші 40 біт): 0100110101111001001000000110111001100001...

✓ Повідомлення приховано в stego.png

✓ Витягнуте повідомлення: My name is Serhii Holosha and I love programming!
✓ Співпадіння: True

--- Аналіз зображення ---
Розмір оригіналу: 120895 байт
Розмір стего: 666821 байт
Різниця: 545926 байт
Змінено пікселів: 117/409600 (0.03%)
Візуальна різниця: непомітна для ока (зміна LSB на ±1)
```

Рис. 17. Виконання програми

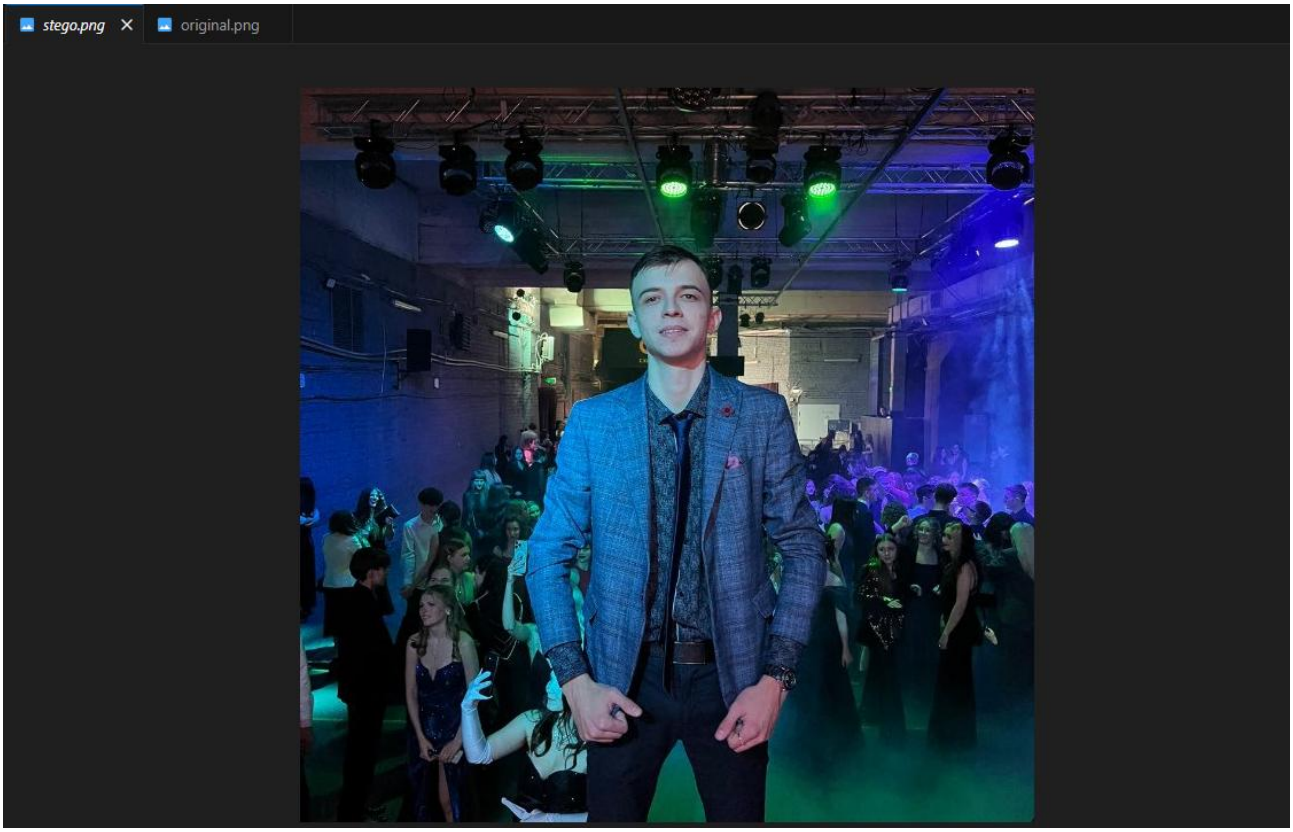

Рис. 18 Створене зображення

Посилання на GitHub: [zahist-informaciji/lab3 at main · serhiikholosha57/zahist-informaciji]

Висновок: в ході лабораторної роботи я оволодів навичками приховування та виявлення інформації в мультимедійних файлах за допомогою стеганографічних методів.