Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення»

Виконав: ст. 2-го курсу, групи 2КІТС-22м
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
_____ Чечелюк О.В

Керівник: к.т.н., доц., доцент каф. МБІС
_____ Карпінець В.В.
«04» грудня 2023 р.

Опонент: к.т.н., доц., доцент каф. ОТ
_____ Колесник І.С.
«04» грудня 2023 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

_____ Юрій ЯРЕМЧУК
"04" грудня 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції/УБ, кафедра МБІС
_____ **Юрій ЯРЕМЧУК**
20 вересня 2023 р.

З А В Д А Н Н Я

на магістерську кваліфікаційну роботу студенту

Чечелюк Олександр Васильович

1. Тема магістерської кваліфікаційної роботи Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення

Керівник магістерської кваліфікаційної роботи Карпінець Василь Васильович, кандидат технічних наук, доцент, завідувач кафедри Менеджменту та безпеки інформаційних систем затверджені наказом вищого навчального закладу від "18" вересня 2023 року № 247

2. Строк подання студентом роботи_____

3. Вихідні дані до МКР: спеціальна література, нормативні документи, Державні стандарти України, наукові доповіді та статті, які стосуються теми магістерської кваліфікаційної роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) у першому розділі – дослідити теоретичні засади захисту цифрового зображення та зробити аналіз існуючих методів захисту;
у другому розділі – дослідити шляхи вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення;
у третьому розділі – програмно реалізувати розроблений алгоритм .

5. Перелік графічного матеріалу:

У першому розділі наведено один рисунок та одна таблиця.

У другому розділі наведено два рисунка та одна таблиця.

У третьому розділі наведено п’ять рисунків та одна таблиця.

У четвертому розділі наведено вісім таблиць

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перший	Карпінець В.В., к.т.н., доц.		
Другий	Карпінець В.В., к.т.н., доц.		
Третій	Карпінець В.В., к.т.н., доц.		
Четвертий	Причепа І.В., к.е.н., доц.		

7.Дата видачі завдання 20 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Визначення напрямку магістерської кваліфікаційної роботи, формулювання теми	20.09.2023	25.09.2023	
2	Аналіз предметної області обраної теми	26.06.2023	30.10.2023	
3	Апробація отриманих результатів	31.10.2023	02.10.2023	
4	Розробка алгоритму роботи	03.10.2023	17.10.2023	
5	Написання магістерської кваліфікаційної роботи на основі розробленої теми	18.10.2023	10.11.2023	
6	Розробка економічної частини	11.11.2023	23.11.2023	
7	Попередній захист магістерської кваліфікаційної роботи	24.11.2023	25.11.2023	
8	Виправлення, уточнення, коригування роботи	26.11.2023	30.11.2023	
9	Захист магістерської кваліфікаційної роботи	15.12.2023	15.12.2023	

Студент _____ Чечелюк О.В.

Керівник роботи _____ к.т.н., доц. Карпінець В. В.

АНОТАЦІЯ

УДК: 004.932.056.5

Чечелюк О. В. Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2023. 98 с.

На укр. мові. Бібліогр.: 40 назв; рис.: 8; табл. 11.

Під час виконання магістерської кваліфікаційної роботи було здійснено вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

В першому розділі було детально розглянуто стеганографічний захист інформації, а саме захист зображення за допомогою ЦВЗ. Проаналізовано основні методи захисту інформації, визначено переваги та недоліки кожного з них. Обрано метод для вдосконалення.

В другому розділі було проаналізовано шляхи вдосконалення обраного методу. Було розроблено алгоритм вдосконалення стеганографічного методу дискретного вейвлет-перетворення на основі колірної моделі HSV. Результати продемонстрували, що запропонована схема має високу здатність до вбудовування та прийнятну непомітність у якості візуального зображення.

У третьому розділі роботи здійснено практичну реалізацію програмного додатку на основі вдосконаленого алгоритму. Результати тестування свідчать, про успішність вдосконаленого методу та доцільність його застосування на практиці.

У четвертому розділі роботи здійснено аналіз економічної доцільності розробки, який свідчить про її високий комерційний потенціал та доцільність подальшого впровадження.

Ключові слова: цифровий водяний знак, HSV, дискретного вейвлет-перетворення

ABSTRACT

Oleksandr Checheliuk. The improvement of information hiding method based on the HSV color model and discrete wavelet transformation. Master's qualification work in specialty 125 – «Cyber Security», Education Program « Cybersecurity of Information Technologies and Systems». Vinnitsa: VNTU, 2023. 98 p.

In Ukrainian. Bibliography: 40 titles; Figures: 8; Table 11

During the master's qualification work, the method of hiding information based on the HSV color model and discrete wavelet transform was improved.

In the first section, we delved into the steganographic protection of information, specifically safeguarding an image using a digital signature. The image is regarded as a vessel for embedding CEV. We scrutinized the primary methods of information protection, evaluating their respective pros and cons, and chose an enhancement method.

The second section assesses avenues for enhancing the selected method and identifies the most pertinent one. An algorithm for refining the steganographic approach using discrete wavelet transform based on the HSV color model was devised. Experimental results attest to the proposed scheme's robust embedding capacity and its reasonable invisibility within the visual image.

In the third section of the paper, a practical implementation of a software application founded on the enhanced algorithm is executed. Test results validate the success of the improved method and its practical applicability.

The fourth section of the paper scrutinizes the economic viability of the development, indicating its significant commercial potential and the feasibility of further implementation.

Keywords: digital watermark, HSV, discrete wavelet transformation.

ЗМІСТ

ВСТУП

Актуальність теми. В сучасному цифровому світі захист конфіденційної інформації є критично важливим завданням. Інформаційна безпека є пріоритетом у багатьох сферах, включаючи фінанси, медицину, військовий сектор та комерційні додатки. Покращення методів приховування даних може відкривати нові можливості для захисту конфіденційності.

Цифровий водяний знак (ЦВЗ) являє собою дані, що впроваджуються в інформаційний об'єкт з метою контролю його використання. Технологія ЦВЗ заснована на застосуванні стеганографічних прийомів, у рамках яких приховується факт наявності ЦВЗ в інформаційному об'єкті.

Отже, обрана тема є актуальною на сьогоднішній день, адже захист авторських прав та прав власності на цифрові зображення дає можливість власникові бути впевненим у їх цілісності та захищеності. А збереження основних характеристик зображення під час вбудовування ЦВЗ забезпечує приховування самого факту його вбудовування. Саме тому завжди буде актуальним вдосконалення методів стеганографічного вбудовування ЦВЗ в цифрове зображення.

Мета і задачі дослідження. Метою дипломної роботи є вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення шляхом розширення областей вбудовування.

Задачами дослідження є:

- аналіз стеганографічного методу дискретного вейвлет-перетворення на основі колірної моделі HSV;
- визначення напрямків вдосконалення стеганографічного методу дискретного вейвлет-перетворення;
- розробка алгоритму роботи програмного засобу на основі вдосконаленого методу;

- проектування та розробка інтерфейсу користувача та реалізація програмного засобу;
- тестування розробки та аналіз отриманих результатів;
- розробки. економічне обґрунтування доцільності впровадження здійсненої розробки на основі вдосконаленого методу вбудовування ЦВЗ.

Об’єкт дослідження захист зображення за допомогою вбудовування в нього цифрового водяного знаку.

Предметом дослідження є стеганографічний метод дискретного вейвлет-перетворення призначений для захисту зображень шляхом вбудовування в нього ЦВЗ.

Новизна роботи: вдосконалення методу вбудовування ЦВЗ на основі алгоритму DWT шляхом розширення областей вбудовування.

Практична цінність: розроблено програмний продукт який реалізує вдосконалений метод вбудовування ЦВЗ.

1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЩОДО ЗАХИСТУ ІНФОРМАЦІЇ

Стеганографія, яка досліджує методи захисту інформації шляхом таємного приховування її існування в різних середовищах, має довгу історію, що налічує тисячоліття. Приховання таємного повідомлення завжди було актуальним для його захисту, а різноманіття технічних, хімічних, фізичних і психологічних методів стеганографії створювало можливості для його успішної реалізації [1].

Навіть при наявності великої кількості відкритих публікацій та річних конференцій, стеганографія тривалий час не мала стабільної термінології. З середини 80-х років минулого століття для опису моделі стеганографічної системи використовувалася "проблема ув’язнених", запропонована G.J. Simmons у 1983 році. Основні поняття стеганографії були узгоджені на 1-й Міжнародній конференції з приховування даних, яка відбулася в 1996 році – Information Workshop on Information Hiding’96. Навіть тоді таке поняття, як "стеганографія", трактується різними фахівцями по-різному. Деякі розуміють під стеганографією лише приховану передачу інформації, тоді як інші включають до неї такі застосування, як метеорний радіозв’язок, радіозв’язок із псевдовипадковим перестроюванням частоти, широкосмуговий радіозв’язок.

Перша конференція, присвячена стеганографії, відбулася в липні 2002 року. На сьогодні існує велика кількість програмної реалізації відомих алгоритмів, проте відчутна відсутність програм початкового рівня, які б крок за кроком демонстрували весь процес стеганографічного перетворення. Такі програми могли б бути корисними в навчальному процесі під час підготовки фахівців у сфері захисту інформації.

У цьому розділі ми розглянемо теоретичний матеріал та дослідження в області стеганографічного захисту інформації, оцінимо актуальність цієї теми, визначимо характеристики та структуру цифрових водяних знаків як засобу

захисту інформації, проведемо аналіз алгоритмів їх генерування та вбудовування, а також розглянемо атаки на системи цифрових водяних знаків, що підкреслює актуальність вдосконалення методів вбудовування цифрових водяних знаків.

1.1. Основні поняття

У своїй роботі з систем забезпечення секретності, Шеннон висловив ідею, що системи приховування інформації є перш за все психологічною проблемою. Давайте розглянемо ситуацію двох в'язнів як модель. Аліса і Боб намагаються обмінюватися повідомленнями, але їхні комунікації постійно перехоплюються начальником в'язниці, Венді. Якщо Венді вважає повідомлення, якими обмінюються Аліса і Боб, підозрілими, вона зупинить їхню комунікацію. Ця модель може застосовуватися в реальних сценаріях, таких як деспотичні режими або урядові політики, що обмежують використання криптографії в межах країни. Таким чином, бажання до конфіденційності між сторонами породжує необхідність в схемах приховування інформації так, щоб спостерігач не міг розрізнити секретне повідомлення, передане між сторонами, та звичайне повідомлення (частину їхньої розмови). У цьому суть базової проблеми класичної стеганографії [2].

Стеганографія – це наука і мистецтво зберігання та передавання секретних повідомлень прихованими каналами всередині відкритих каналів передавання так, що факт передавання секретних даних залишається невідомим для неавторизованого користувача.

Загальні підходи:

- Повне приховання каналу передавання;
- Ускладнення виявлення, отримання та модифікації повідомлень, прихованих всередині відкритих даних;
- Маскування секретних повідомлень, використовуючи протоколи.

Методи цифрової та комп'ютерної стеганографії стали ефективними засобами для вирішення завдань, пов'язаних із створенням різноманітних

систем контролю за дотриманням авторських прав на ринку цифрової фото-, аудіо-, відеопродукції та поширення програмних продуктів (водяні знаки, підписи, захист від підробки, вставка заголовків в оцифровані аналогові сигнали і таке інше). Цей факт є значущим стимулом для фінансування досліджень у цій області з боку великих виробників та некомерційних організацій. Зокрема, WatermarkingWorld представляє собою міжнародну асамблею фахівців зі стеганографії, які розробляють системи вбудовування цифрових водяних знаків [4]. Використання методів стеганографії надзвичайно доцільне в тих країнах, де використання криптографічних засобів заборонено.

Основними поняттями стеганографії є:

- Контейнер b (container, carrier) – відкриті дані, які використовуються для приховання секретної інформації/в які вбудовується секретне повідомлення;
- Повідомлення m (message, payload) – секретне повідомлення, яке необхідно приховати;
- Ключ k (key) – секретна інформація, що відома тільки авторизованому користувачу та визначає конкретний алгоритм вбудовування;
- Пустий контейнер c (emptycontainer, unmodifiedcontainer) – контейнер являє собою послідовність елементів довжини l_c;
- Модифікований контейнер або стеганограма s (modifiedcontainer, package, steganogram) – контейнер, який містить секретне повідомлення;
- Стеганографічний алгоритм являє собою два перетворення, пряме (E_K: Mx C xK → B) та зворотне (D_K: C xK → M) [4];
- Стеганографічна система або стеганосистема (steganographicsystem, steganosystem) – сукупність повідомлень, секретних ключів, контейнерів та перетворень, що їх об’єднують: $\Psi = (C, M, K, D_K, E_K)$, де C є сукупністю усіх контейнерів, M та K – повідомлень та ключів відповідно, так що $D_K(E_K(c, m, k), k) = m$.
- Пакування повідомлення до контейнера відбувається за допомогою стеганокодера;

- Стеганографічний канал – канал передачі контейнера-результату;
- В стеганодетекторі здійснюється визначення наявності в контейнері (можливо вже зміненому) прихованих даних. Ця зміна може бути обумовлена впливом помилок в каналі зв’язку, операцій обробки сигналу, навмисних атак порушників. В багатьох моделях стеганосистем сигнал-контейнер розглядається як адитивний шум. Тоді задача виявлення і виділення стеганоповідомлення є класичною для теорії зв’язку. Але такий підхід не враховує двох факторів: невипадкового характеру контейнера і вимог по збереженню його якості [5].

На рисунку 1.1 зображена блок-схема, яка відображає основні компоненти стеганосистеми та їх взаємодію. Важливо відзначити, що ця блок-схема розглядає сценарій лише з пасивним порушником (Венді не може модифікувати повідомлення, а тільки дозволяє чи забороняє його передачу; порушник, який вносить зміни у канал, контейнер або повідомлення, вважається активним). У більшості випадків важливим є одночасне дотримання умов приховування даних при пасивній та активній протидії [6].

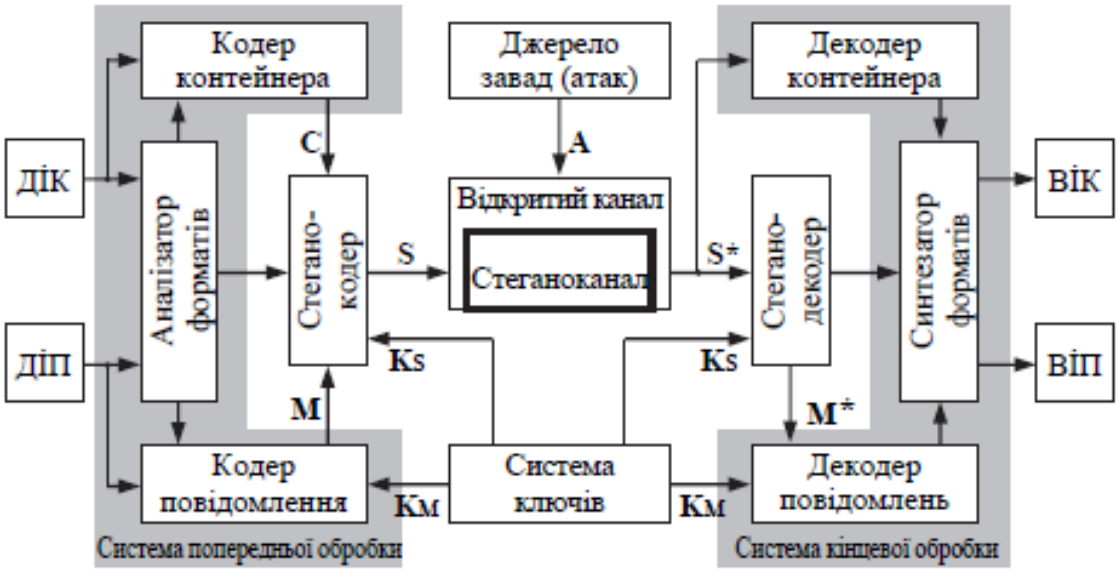


Рисунок 1.1. Компоненти типової стеганосистеми та їх взаємодія

На рисунку 1.1 подається сигнал з виходу джерела інформації-контейнера (ДІК), який має бути переданий системою зв’язку відкритим

чином і може мати визначений формат. Цей сигнал подається на входи аналізатора форматів (АФ) та кодера контейнера у системі попередньої обробки (СПО). Крім того, на входи АФ та кодера повідомлення СПО надходить сигнал з виходу джерела інформації-повідомлення (ДІП), який також може мати різний формат. АФ, визначаючи структуру інформації на своїх входах, генерує керувальні сигнали для відповідних кодерів щодо необхідності застосування операцій додаткового кодування (перекодування) або його відсутності. Кодери контейнера і повідомлення взаємодіють для забезпечення сумісності форматів ДІК та ДІП з інструментами подальшої цифрової обробки у стеганокодері, де здійснюється вбудовування повідомлення M до контейнера C [7].

Стеганографія використовує контейнери різної природи, вибір яких залежить від конкретного завдання. Комп'ютерна стеганографія, орієнтована на використання комп'ютерної техніки, включає у себе реалізацію стеганосистем з використанням цифрової інформації у вигляді файлів. Таким чином, елементами стеганосистеми є файл-контейнер та файл-повідомлення. Важливо, щоб файл-контейнер не викликав підозри, а додавання секретної інформації не змінювало його основні властивості [8].

Якщо аудіо-файл використовується як контейнер, то довжина елементів буде визначатися кількістю відрахунків за одиницю часу. У випадку, коли контейнером служить цифрове зображення, послідовність елементів можна отримати шляхом векторизації зображення, тобто перетворення масиву пікселів на вектор.

Більшість методів стеганографії основані на двох ключових принципах:

- Людські органи чуттів не можуть розрізнити незначні зміни при сприйнятті кольору, форми та звуку;
- Як наслідок, існують файли, для яких абсолютна точність не обов’язкова, тобто, їх можна модифікувати без втрати їх функціонального значення.

Отже, використані методи передбачають виявлення незначущих фрагментів контейнера та заміщення їх інформацією, яку необхідно приховати [9].

Системи стеганографії можуть бути розділені на ключові та безключові. У цьому аспекті теорія стеганографії має свої особливості. У випадку ключових систем, подібно до криптографії, існують стеганографія із використанням відкритого та закритого ключа (secretkeysteganography, publickeysteganography). У першому випадку модель не розглядає, яким чином відбувається обмін секретним ключем, і часто використовуються підходи, схожі на ті, що в криптографії. Проте є і безключові системи (чиста стеганографія), де не враховується обмін секретним ключем [10].

Зазначимо знову, що основною проблемою класичної стеганографії є передача повідомлення так, щоб третя сторона навіть не підозрювала про наявність прихованого каналу передачі. Таким чином, питання ключа (секретного чи відкритого) тут і не стоїть. Безключові системи в більшій мірі відповідають теоретичним принципам. Однак на практиці ефективність "чистої" стеганосистеми залежить від секретності двох функцій (вбудовування та виділення стеганограми), що суперечить принципу Кірхгоффса – безпека системи повинна залежати тільки від секретності ключа, а не алгоритму.

1.2 Актуальність дослідження обраної галуз

Останні різкі зміни в розвитку інформаційних технологій, які ми спостерігаємо, призвели до того, що значна частина інтелектуальної власності знаходиться і обробляється в комп'ютерних мережах чи циркулює у цифровому форматі. Розвиток обчислювальної техніки зокрема сприяв розвитку комп'ютерної стеганографії.

Мова, відео, аудіозаписи та зображення, хоча властиво мати аналоговий характер, зараз часто представлені у цифровому форматі і часто використовуються для вбудовування прихованих повідомлень. Іноді

інформація приховується в текстових файлах або в виконуваних файлах програм [11].

На сьогоднішній день, основні порушення прав інтелектуальної власності включають піратство, плагіат, підробку інформації, зміну інформації та недобросовісну конкуренцію. Особлива увага приділяється захисту прав інтелектуальної власності, яка розповсюджується у цифровому вигляді через цифрові носії та мережу Інтернет.

Між завданнями, які вирішуються в рамках систем захисту, важливе місце відводиться завданню спеціального кодування інформації як даних, призначених для прихованої передачі [12].

Розробка стеганографічних методів привертає увагу багатьох фахівців, які працюють над розробкою нових технологій для забезпечення високої надійності інформаційних систем. Загалом, завдання стеганографії та стегоаналізу є однією з ключових проблем у теорії безпеки та надійності інформаційних технологій. На відміну від криптографії, яка обмежує доступ до інформації, що міститься в повідомленні та передається за допомогою секретного ключа, стеганографія має на меті приховати факт передачі будь-якого повідомлення від сторонніх осіб. Зазвичай це вирішується вбудовуванням секретного повідомлення у безпечний об'єкт даних, відомий як контейнер, так, щоб факти існування або передачі не викликали підозри [13].

У цифровій стеганографії [14], як контейнер використовується цифровий об'єкт, зазвичай комп'ютерний файл. Сучасні методи вбудовування дозволяють приховувати інформацію у файли аудіо, відео, тексту, виконувані програми та інші. На ринку існує велика кількість програм для стеганографії, як комерційних, так і безкоштовних, з графічним інтерфейсом або у вигляді консольних додатків.

Основні характеристики методів стеганографії включають обсяг вбудовуваного повідомлення та стійкість до аналізу (виявлення факту вбудовування). Технологія цифрового водяного знаку також є важливою в цьому контексті, зокрема для захисту від копіювання, прихованої анотації

документів, доказу автентичності інформації та створення прихованого зв'язку.

Цифрова стеганографія широко використовується для захисту авторських прав. За допомогою вбудованих міток або "відбитків пальця" можна ідентифікувати законного користувача. Наприклад, програми можуть супроводжуватися мітками, що ідентифікують ліцензійних покупців, полегшуючи виявлення піратських копій [15].

Популярність досліджень у цій галузі обумовлена обмеженнями деяких країн на використання криптографічних методів захисту інформації, а також необхідністю враховувати права власності на цифрову інформацію. Це призводить до інтенсивних досліджень у сфері стеганографії, яка займається приховуванням повідомлень і використанням цифрових водяних знаків.

Отже, враховуючи суттєві переваги та особливості застосування стеганографічного методу захисту інформації, подальше дослідження в цій області є актуальним і вимагає більш детального розгляду.

1.3 Порівняння існуючих стеганографічних методів

Більшість сучасних досліджень у стеганографії фокусуються на використанні цифрових зображень як стегоконтейнерів. Це обумовлено кількома причинами:

- Необхідність захисту цифрових фотографій та відео від незаконного поширення.
- Великий обсяг цифрових зображень, що дозволяє вбудовувати велику кількість інформації або підвищувати стійкість вбудовування.
- Відомість розміру стегоконтейнера та відсутність обмежень реального часу для приховування інформації.
- Наявність на більшості реальних зображень областей з шумовою структурою, що добре підходять для вбудовування інформації.

- Обмежена спроможність людського зору розрізняти незначні зміни в кольорах, яскравості, контрастності, наявність шуму та зміни біля контурів зображення.
- Розвиток методів цифрової обробки зображень.

Комп'ютерна стеганографія базується на двох принципах:

- По-перше, аудіо- та відеофайли, а також файли з оцифрованими зображеннями можна деякою мірою змінити без значного спотворення.
- По-друге, можливості людини розрізняти дрібні зміни кольору або звуку обмежені.

Базуючись на цьому було сформовано основні методи приховування зображень в стеганографії [16]:

- Методи заміни. Загальний принцип даних методів полягає в заміні надлишкової, малозначущої частини зображення бітами секретного повідомлення. Для вилучення повідомлення необхідно знати місце, де була розміщена приховувана інформація.
- Методи приховування у частотній області зображення – це техніки, за допомогою яких інформація приховується або вбудовується в частотний спектр цифрового зображення. Основна ідея полягає в тому, що зображення представляється у вигляді послідовності частотних складових (наприклад, DCT-коефіцієнтів для JPEG-зображень), і інформацію вбудовують у ці складові таким чином, щоб вона була малопомітною для звичайного спостерігача.
- Широкосмугові методи. Суть методів полягає в значному розширенні смуги частот сигналу, більш ніж це необхідно для передачі реальної інформації. Розширення діапазону виконується в основному за допомогою коду, який не залежить від переданих даних. Корисна інформація розподіляється по всьому діапазону, тому при втраті сигналу в деяких смугах частот в інших смугах присутньо достатньо інформації для її відновлення.
- Статистичні методи приховують інформацію шляхом зміни деяких статистичних властивостей зображення. Вони засновані на перевірці

статистичних гіпотез. Суть методу полягає в такій зміні деяких статистичних характеристик контейнера, при якому одержувач зможе відрізнити модифіковане зображення від немодифікованого.

- Структурні та методи спотворення інформації. Методи спотворення, на відміну від попередніх методів, вимагають знання про первісний вигляд контейнера. Схема приховування полягає в послідовному проведенні ряду модифікацій контейнера, які вибираються відповідно до секретного повідомлення. Для вилучення прихованих даних необхідно визначити всі відмінності між стеганограмою і вихідним контейнером.
- Кожен із цих методів має свої особливості і використовується в різних ситуаціях в залежності від конкретних вимог та цілей стеганографії.

1.4 Аналіз застосування цифрової стеганографії

Цифрова стеганографія – це розділ стеганографії, що вивчає надійне приховування певних бітових послідовностей в стеганоконтейнерах (фото, документи, аудіо-файли). В такій системі непомітність передбачає включення людини в стегосистему, а надійність – це стійкість до спотворень різних видів. В такій системі людина розглядається як додатковий приймач інформації, що може пропонувати до стегосистеми важко формалізовані вимоги. Разом з тим, напрямок стеганографії як засіб захисту інформації має широке застосування у багатьох галузях (табл. 1.1.).

Таблиця 1.1 Галузі застосування стеганографії

Мета захисту	Галузь застосування
Захист від копіювання	Електронна комерція, контроль за копіюванням DVD, розповсюдження мультимедійної інформації
Прихована анотація документів	Медичні знімки, картографія, мультимедійні база даних
Аутентифікація	Системи відеоспостереження, електронні комерції, голосова пошта, електронне конфіденційне діловодство
Прихований зв'язок	Воєнні та розвідувальні додатки, а також застосування у випадках, коли неможливе застосування криптографічних методів

За останні декілька років стеганографія набуває все більшого поширення у різних сферах життя людини, у зв’язку з цим сформовано такі її напрями:

- вбудовування інформації з метою її передачі;
- вбудовування цифрових водяних знаків;
- вбудовування ідентифікаційних номерів;
- вбудовування заголовків.

Основні переваги цифрової стеганографії полягають у тому, що вона забезпечує конфіденційність, оскільки вбудована інформація може бути прихована в невеликих змінах, які важко помітити без спеціального аналізу. Крім того, відсутність видимих змін в оригінальному файлі гарантує незалежність від детекторів інших методів аналізу, таких як антивірусні програми.

Важливо враховувати, що стеганографія не забезпечує шифрування даних; вона лише приховує наявність додаткової інформації, але не забезпечує захист від розшифрування або злому. Крім того, деякі методи стеганографії можуть впливати на якість файлу, особливо при вбудовуванні значних обсягів даних.

Цифровий водяний знак являє собою дані, що впроваджуються в інформаційний об'єкт з метою контролю його використання. Технологія ЦВЗ заснована на застосуванні стеганографічних прийомів, у рамках яких приховується факт наявності ЦВЗ в інформаційному об'єкті (контейнері). Проте, в ньому зберігатиметься інформація, що може бути зчитана з даного контейнера при наявності відповідного стеганографічного ключа, що визначатиме права доступу до елементів цифрового водяного знаку. Сьогодні, цифрові водяні знаки активно застосовуються для здійснення контролю за використанням мультимедійного контенту, електронних документів. Суттєва особливість файлів-контейнерів мультимедійного контенту полягає в тому, що всі вони є пасивними інформаційними об'єктами, що виконують тільки функцію зберігання даних. Очевидно, що необхідність у контролі використання інформаційних об'єктів не обмежується тільки контейнерами

цього виду. Така необхідність має місце і для активних інформаційних об'єктів які виконують деяку обчислювальну або керівну функцію.

1.5 Висновки та постановка задач

Таким чином в даному розділі було представлено поняття стеганографії, основні визначення, параметри, об’єкти та суб’єкти стеганографічних систем. Було виконано огляд існуючих популярних методів стеганографії, а також перспективних напрямків її розвитку. Проаналізовано основні методи захисту інформації, визначено переваги та недоліки кожного з них. Також через ряд перевагу було обрано метод дискретного вейвлет-перетворення.

Виходячи із отриманих результатів аналізу, далі для виконання роботи поставлені такі задачі:

- здійснити вдосконалення методу приховування інформації на основі;
- розробити алгоритм роботи програмного засобу на основі вдосконаленого методу;
- здійснити проектування та розробку інтерфейсу користувача, а також реалізацію програмного засобу;
- здійснити тестування розробки та аналіз отриманих результатів;
- економічно обґрунтувати доцільність впровадження здійсненої розробки на практиці на основі вдосконаленого методу.

Виконання поставлених задач дозволить реалізувати основну мету даної роботи, а саме здійснити методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

2. ВДОСКОНАЛЕННЯ МЕТОДУ ПРИХОВУВАННЯ ІНФОРМАЦІЇ НА ОСНОВІ КОЛІРНОЇ МОДЕЛІ HSV ТА ДИСКРЕТНОГО ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ

В даному розділі описано вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення. В ході написання розділу обґрунтованого запропоноване удосконалення, наведено алгоритми вбудовування та вилучення даних із стегоконтейнера, описано алгоритм роботи програмної розробки на основі удосконаленого методу, а також здійснено вибір програмних засобів для подальшої практичної реалізації програмного додатку.

2.1. Аналіз шляхів удосконалення обраного методу

Незалежно від алгоритму обробки та задіяного апаратного чи програмного забезпечення, цифрові зображення в будь-якому випадку сприймаються зоровою системою людини, як і будь-який інший елемент навколишнього середовища. На завершальному етапі обробки вони виводяться на периферійні пристрої, такі як монітори, дисплеї смартфонів, планшетів [17].

Стеганографічні системи діють, вводячи в оману системи сприйняття, включаючи зір. Таким чином, для розуміння подальшого матеріалу та запропонованого методу необхідно розглянути зорову систему людини.

Зорове сприйняття забезпечується органами зору (очі) та зоровим аналізатором. Зорові функції включають світловідчуття, кольорове відчуття, центральний та периферичний зір, а також стереоскопія.

Світловідчуття – це здатність сприймати світло як специфічний подразник в діапазоні сонячного випромінювання і пристосовуватися до сприйняття навколишнього середовища при різних рівнях освітленості.

Світловідчуття людини має такі кількісні характеристики [18]:

- Межа подразнення – це визначається мінімальною кількістю світлової енергії, яка викликає відчуття світла.
- Світлочутливість – це величина, зворотна межі подразнення, тобто як чутлива людина до світла.
- Межа розрізнення – це визначається мінімальною помітною для ока різницею в освітленості.
- Швидкість світлової адаптації – це те, наскільки швидко очі адаптуються до зміни рівня освітленості.
- Швидкість темнової адаптації – це те, наскільки швидко очі адаптуються до темряви після перебування в освітленому середовищі.

Пристосування до змін освітленості відоме як адаптація, де світлова адаптація відбувається у яскравому світлі, а темнова адаптація – у слабкому освітленні.

Кольорове відчуття включає здатність розрізняти хвилі світла за їхньою довжиною. Кожен колір характеризується тоном, насиченістю та яскравістю. Тон визначається довжиною хвилі спектрального кольору, насиченість пов'язана з вмістом цього тону, а яскравість залежить від домішок білого. Різноманіття кольорових відтінків виникає зі змішування семи чистих тонів спектра: червоного, зеленого та синього [19].

Центральний або предметний зір відповідає за розпізнавання форми та розміру предметів, базуючись на роздільній здатності точок, які обмежують кожен предмет у просторі.

Периферичний зір – функція, що забезпечує просторове орієнтування.

Стереоскопія – здатність сприймати об’єм та відстань між об’єктами, а також між спостерігачем та об’єктом у спокої та у русі [20].

Людське око сприймає колірну інформацію в діапазоні хвиль приблизно від 380 нм (синій колір) до 770 нм (червоний колір), причому найкращу чутливість має в діапазоні 510 – 530 нм (зелений колір). При цьому, згідно із законом тривимірності сприйняття кольору, око реагує саме на ці три складові – червону, зелену та синю.

Для практичного застосування стеганографії важливими параметрами є пороги кольоророзрізнення, тобто мінімальна зміна кольоровості, яку здатне помітити людське око. Кольоророзрізнення має нелінійний характер [21].

На основі кількісних характеристик, які використовуються для математичного опису зорової системи, виділяють її властивості, спершу об’єднані у дві групи – низькорівневі (фізіологічні) та високорівневі (психофізіологічні). Різниця між ними полягає у тому, що високорівневі властивості проявляються не одразу зі сприйняттям хвиль видимого спектру, а вже після обробки інформації мозком та видачі ним команд на «підлаштування» зору. До низькорівневих властивостей належать (основні, які впливають на розрізнення спотворення у зображенні) [22]:

- Чутливість до зміни яскравості – виявлення зміни яскравості при подоланні порогу адаптації; наразі межа нерозрізненості оцінюється у 0,01-0,03 від середніх значень яскравості;
- Частотна чутливість – через нерівномірну амплітудно-частотну характеристику зорової системи, людина є більш сприйнятливою до низькочастотного, ніж до високочастотного шуму;
- Ефект маскування – збудження одних і тих самих підканалів зорової системи компонентами, що мають подібні характеристики, призводить до підвищення порогу виявлення сигналу в присутності іншого сигналу із аналогічними характеристиками; так, наприклад, адитивний шум менш помітний на високочастотних ділянках зображення.

До високорівневих властивостей відносять [23]:

- Чутливість до контрасту. Особливу увагу при сприйнятті звертають висококонтрасні ділянки зображення, де спостерігаються інтенсивні перепади яскравості.
- Чутливість до розміру деталей. Великі ділянки зображення вражають більше, аніж менші за розміром, з урахуванням наявності межі насичення, коли подальше збільшення розміру не має значущого впливу.

- Чутливість до форми. Довгі та тонкі об'єкти привертають більше уваги порівняно з круглими однорідними об'єктами.
- Чутливість до кольору. Деякі кольори, особливо червоний, виділяються більше, зокрема при відмінності фону від кольору об'єктів.
- Чутливість до місця розташування. Спостерігач схильний першочергово розглядати центр зображення та передній план, фокусуючись на обличчях, людях, очах та руках.
- Чутливість до зовнішніх подразників. Рух очей спостерігача залежить від конкретних умов, отриманих інструкцій та додаткової інформації в конкретній обстановці.

Усі ці властивості враховуються при обробці цифрових зображень та є ключовими для розробки стеганографічних систем, особливо тих, які використовують візуальні контейнери.

2.2 Вибір частотного перетворення для приховування інформації

Як уже зазначалося, стеганографічні методи заміни нестійкі до будь-яких спотворень, а застосування операції стиснення із втратами призводить до повного знищення всієї секретної інформації, прихованої методом метод заміни найменшого значущого біта в зображенні. Більш стійкими до різних спотворень, у тому числі і стиснення, є методи, які використовують для приховування даних не тимчасову область, а частотну [24].

Існує кілька способів представлення зображення в частотній області. Наприклад, з використанням дискретного косинусного перетворення (ДКП), дискретного перетворення Фур’є (ДПФ), перетворення Карунена-Лоева або вейвлет-перетворення [25]. Дані перетворення можуть застосовуватися як до всього зображення, так і до деяких його частин.

Основна ідея ДКП полягає в тому, щоб представити сигнал або послідовність даних як лінійну комбінацію косинусних функцій різних частот. Для дискретного сигналу ДКП визначається наступним чином:

$$X_k = \sum_{i=0}^{N-1} x_n \, COS\left(\frac{(2n+1)k\pi}{2N}\right), \qquad (2.1)$$

де x_n– вхідний сигнал, N – кількість відліків, а X_k – коефіцієнти ДКП.

Важливо відзначити, що ДКП здійснює перетворення з просторової області у частотну, де кожен коефіцієнт відображає важливість певної частоти в сигналі. Низькі частоти відображаються в перших коефіцієнтах, тоді як високі частоти – в більших номерах коефіцієнтів [26].

Дискретне перетворення Фур'є – це математичний метод, який використовується для перетворення дискретної послідовності чисел з часової області в частотну область [27]. ДПФ розкладає послідовність на суму синусоїд та косинусоїд різних частот, що дозволяє аналізувати складові частоти сигналу.

ДПФ визначається наступним чином:

$$X_k = \sum_{i=0}^{N-1} x_n \, e^{-j2\pi kn/N}, \tag{2.2}$$

де x_n·– вхідний сигнал, N – кількість відліків, j – уявна одиниця , e – число Ейлера, а X_k – коефіцієнти ДПФ.

Перетворення Карунена-Лоева (КЛ-перетворення) є методом аналізу залежності між двома наборами даних. Основна ідея полягає в тому, щоб проектувати вихідні дані на такі нові вектори, де кореляція між ними буде максимальною. Цей метод часто використовується для виявлення прихованих залежностей або патернів в даних [28].

Дискретне вейвлет-перетворення є математичним методом обробки сигналів, який дозволяє розкласти сигнал на компоненти різної частоти та часової роздільної здатності. Основна ідея вейвлет-перетворення полягає в застосуванні функцій вейвлету для аналізу сигналів [29].

Нехай $x(n)$ – вихідний сигнал, а $\psi_{j,k}(n)$ – вейвлет-функція зсунута на b та масштабована на a [30]. Тоді коефіцієнти ДВП можуть бути обчислені наступним чином:

$$W(a,b) = \sum_n x[n] \cdot \psi^* \left(\frac{n-a}{a}\right), \tag{2.3}$$

Ці коефіцієнти використовуються для рекурсивного розкладання сигналу на апроксимаційні та детальні складові на різних рівнях. Процес може бути інтерпретований як фільтрація та піддискретизація сигналу. Для відновлення сигналу застосовується зворотна процедура, яка використовує апроксимаційні та детальні коефіцієнти [31].

В данні роботі було обрано використання методу дискретного вейвлет-перетворення (ДВП) через ряд переваг описаних нижче :

- Мультирезолюційність: ДВП дозволяє розкласти зображення на різні рівні деталей, включаючи високочастотні і низькочастотні компоненти. Це дозволяє вбудовувати таємну інформацію на різних рівнях деталей, що робить стеганографічну інформацію більш стійкою і непомітною.
- Компактність та ефективність обчислень: ДВП забезпечує компактне подання зображення в термінах коефіцієнтів вейвлет-перетворення, що спрощує процес вбудовування та видобування інформації. Крім того, це дозволяє зберігати більше інформації на мінімальному обсязі.
- Збереження стійкості до стиснення та обробки: ДВП зазвичай зберігає інформацію після стиснення та обробки зображення, що робить стеганографічну інформацію більш стійкою та менш схильною до втрати під час передачі чи зберігання зображень.
- Адаптивність: ДВП може бути адаптований до різних потреб стеганографії, включаючи приховування текстової, аудіо або відео інформації. Різні вейвлети та рівні деталей можуть бути вибрані в залежності від завдання.
- Можливість визначення власних параметрів: ДВП дозволяє налаштувати параметри вбудовування та видобування інформації, включаючи рівні деталей, вейвлет-функції, інтенсивність вбудовування, інтервали для прихованої інформації та інші параметри.

З використанням ДВП в стеганографії можна досягти високого рівня стійкості та ефективності, забезпечуючи при цьому непомітність та надійність прихованої інформації в зображеннях.

Запропонований метод представляє спосіб вбудовування секретних даних в зображення шкіри, оскільки він не так чутливий для візуальної системи людини (HSV) [32]. Він використовує переваги біометричних характеристик, таких як відтінок шкіри, замість вбудовування даних в будь-яку частину зображення. Суть методу можна описати наступним чином: спочатку на вхідному зображенні визначається відтінок шкіри, використовуючи кольоровий простір HSV (відтінок, насиченість, значення). Потім зображення обкладинки трансформується в частотній області, застосовуючи простий перетворювач Хаара, що розбиває зображення на чотири піддіапазони. Після цього визначається корисне навантаження (кількість бітів, які можна приховати в даних). Нарешті, секретні дані вбудовуються в один із високочастотних підсмуг шляхом заміни пікселів шкірного зображення. Перед усіма цими етапами проводиться обрізання вхідного зображення, і вбудовування виконується лише в обрізаній області, а не в усьому зображенні. Обрізання забезпечує більшу безпеку, оскільки обрізана область діє як ключ для декодування [12].

Отже, виходячи із описаного вище, приховування даних у зображеннях на основі дискретного вейвлет-перетворення представити наступними кроками.

Нехай C – оригінальне 24-бітне кольорове зображення обкладинки розміром M × N. Воно позначається наступним чином:

$$C = (x_{ij}, y_{ij}, z_{ij} \mid 1 \leq \mathrm{i} \leq \mathrm{M}, 1 \leq \mathrm{j} \leq \mathrm{N}, x_{ij}, y_{ij}, z_{ij} \in (0, 1..255)) \quad (2.4)$$

Нехай розмір обрізаного зображення дорівнює Mc × Nc, де Mc ≤ M, Nc ≤ N і Mc = Nc, тобто обрізана область повинна бути точно квадратною, оскільки пізніше ми повинні застосувати ДВП до цієї області.

Нехай S – секретні дані. Тут під секретними даними розуміється бінарне зображення розміром a × b. На Рисунок 2.1 зображено блок-схему процесу вбудовування секретних даних.

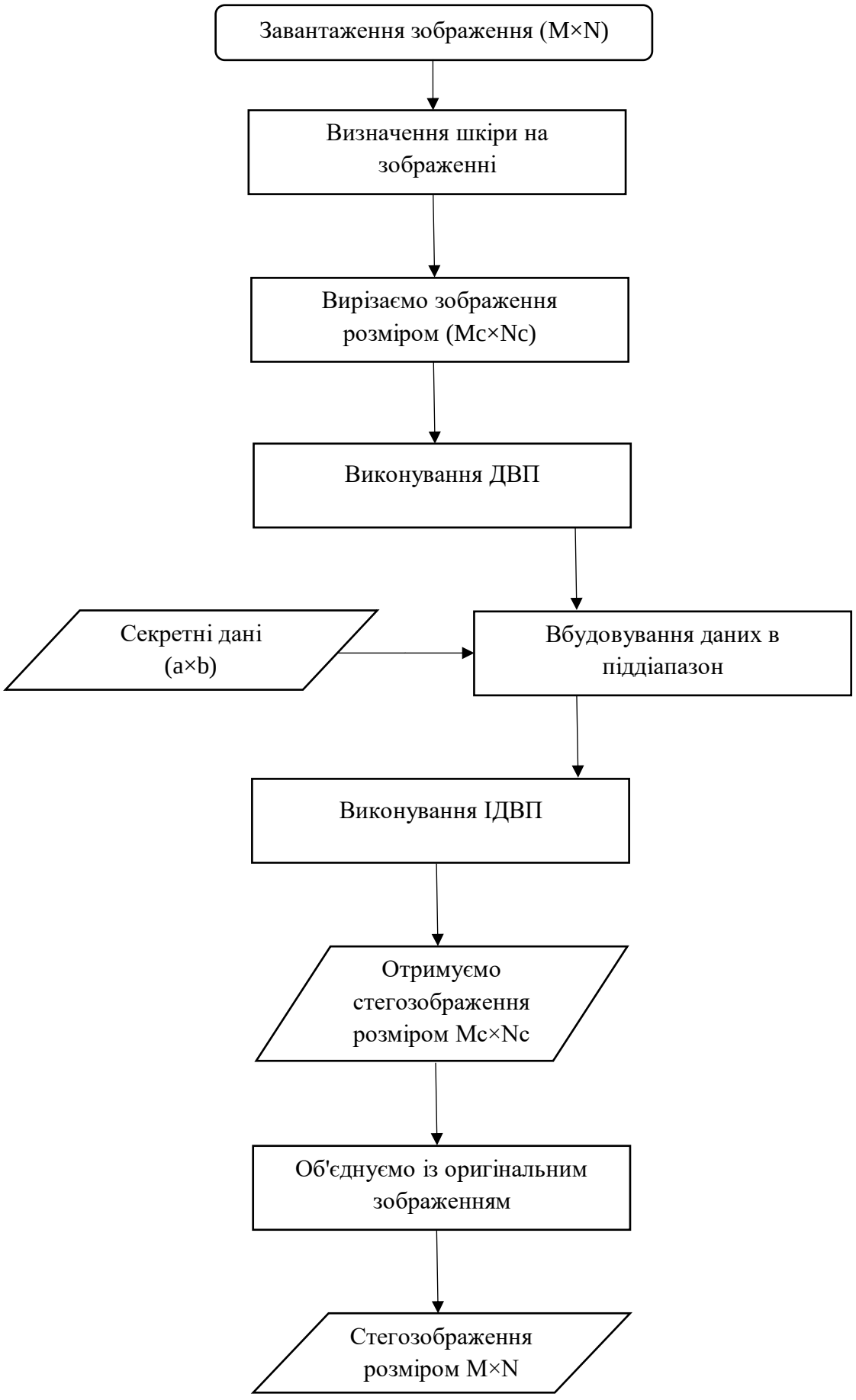


Рисунок 2.1. Блок-схема процесу вбудовування секретних даних

Нижче детально описано різні кроки цієї блок-схеми:

Крок 1: Після завантаження зображення, виконується аналіз визначення людської шкіри. Це призведе до створення зображень, котрі містить чи не містять пікселі людської шкіри.

Крок 2: Попросіть користувача обрізати зображення (Mc×Nc). Після цього оригінальне зображення також буде обрізано по тій самій ділянці. Обрізана область повинна мати точну квадратну форму, оскільки пізніше ми повинні виконати ДВП Хаара, і обрізана область повинна містити ділянки шкіри, такі як обличчя, руки тощо, оскільки ми будемо приховувати дані в пікселях шкіри в одному з піддіапазонів ДВП. Тут обрізання виконується з міркувань безпеки. Обрізаний квадрат буде діяти як ключ на приймаючій стороні. Якщо вона його знає, то можливе лише зчитування даних. Зловмисник може спробувати виконати ДВП на всьому зображенні; в такому випадку атака буде невдалою, оскільки ми застосовуємо ДВП тільки до певної обрізаної області.

Крок 3: Застосовуємо ДВП лише до обрізаної області (Mc×Nc), а не до зображення до усього зображення. В результаті отримуємо 4 піддіапазони, позначені як HLL, HHL, HLH і HHH (всі 4 піддіапазони мають однаковий розмір Mc/2, Nc/2). Корисне навантаження зображення для зберігання секретних даних визначається на основі кількості пікселів шкіри, присутніх в одному з високочастотних піддіапазонів, в якому будуть приховані дані.

Крок 4: Вбудовуємо секретні дані в один з піддіапазонів, який ми отримали раніше, відстежуючи пікселі шкіри в цьому піддіапазоні. Окрім низькочастотного піддіапазону LL, для вбудовування може бути обраний будь-який високочастотний піддіапазон, оскільки піддіапазон LL містить важливу інформацію. Вбудовування в піддіапазон LL сильно впливає на якість зображення. Ми обрали високочастотний піддіапазон HH. Під час трансформації секретні дані будуть вбудовані не в усі пікселі піддіапазону ДВП, а лише в ті пікселі, які є пікселями шкіри. Вбудовування виконується

відповідно до порядку растрового сканування, який вбудовує секретні дані у вибраному піддіапазоні [13], якщо коефіцієнт є пікселем шкіри.

Крок 5: Виконайте інверсне дискретне вейвлет-перетворення (ІДВП), щоб об'єднати 4 піддіапазони.

Крок 6: Отримуємо обрізане стегозображення розміром Mc×Nc отримане на попередньому кроці (крок 5). Після візуального огляду воно має бути схожим на оригінал після візуального огляду, але на цьому етапі воно має розмір Mc×Nc. Nc, тому нам потрібно об'єднати обрізане стегозображення з оригінальним зображенням, щоб отримати стегозображення розміром M×N.

Процедура визначення людської шкіри на зображенні зазвичай перетворює заданий піксель у відповідний колірний простір, а потім використовує класифікатор шкіри, щоб позначити піксель котрі містять або не містять шкіру. Класифікатор шкіри визначає межу розпізнавання класу кольору шкіри в колірному просторі. Хоча цей процес є досить простим, він виявився досить складним завданням. Тому важливим завданням при виявленні шкіри є представлення кольору у способі, який є інваріантним або, принаймні, нечутливим до змін освітлення.

Ще однією проблемою є те, що багато об'єктів у реальному світі можуть мати кольори, схожі на кольори людської шкіри. Це призводить до того, що будь-який детектор шкіри може мати багато хибних спрацьовувань у фоновому режимі, якщо навколишнє середовище не контролюється.

Найпростіший спосіб вирішити, чи є піксель кольором шкіри – це явно визначити межу. RGB-матриця заданого кольорового зображення може бути перетворена в різні колірні простори, щоб отримати області кольору шкіри або кольору, близького до нього. Існує декілька палітр, і в біометричній літературі використовуються два види: HSV (відтінок, насиченість і значення) і YCbCr (жовтий, хроматичний синій, хроматичний червоний). Експериментально встановлено і теоретично доведено, що розподіл кольору шкіри людини постійно перебуває в певному діапазоні в межах цих двох колірних просторів.

В даній реалізації визначено, що колір підшкірної плоті може бути апроксимований сектором шестикутника з обмеженнями:

$$S_{min} = 0.23,$$
$$S_{max} = 0.68,$$
$$H_{min} = 0,$$
$$H_{max} = 50. \quad (2.5)$$

2.3 Розробка вдосконаленого методу для приховування інформації

Якщо проаналізувати велику кількість фотографії то можна помітити що окрім людей на них знаходиться велика кількість інших об’єктів котрий маю сталий діапазон кольорів. До таких об’єктів можна віднести:

- Рослини – зазвичай мають зелений відтінок;
- Небо – зазвичай мають блакитний відтінок;
- Пісок на пляжі – зазвичай мають жовтий відтінок;
- Сніг – зазвичай мають білий відтінок;
- Гірські вершини – зазвичай мають сірий відтінок.

Головна ідея вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення полягає в тому щоб розпізнавати інші об’єкти окрім людської шкіри та використовувати їх для приховування секретної інформації. Для цього потрібно описати критерії за котрими будуть розпізнаватися об’єкти. Наступний крок буде трансформація зображенні в простір кольорів HSV та визначення об’єкта за визначеними критеріями. В даному випадку критерії об’єкта буде діапазон кольорів в яких зазвичай зустрічається цей об’єкта на зображеннях.

Нижче в Таблиця 2.1 буде наведена список об’єктів із їхнім діапазоном кольорів в HSV. Значення для діапазонів були взяти із відкритих баз даних OpenCV. Проте не потрібно прив’язуватися лише до цього переліку, так як якщо котрийсь із об’єктів можна описати за допомогою діапазону кольорів то його теж можна використати в нашому покращеному методу.

Таблиця 2.1 Діапазон кольорів для об’єктів

Об’єкт	Нижній межа	Верхня межа
Рослини	30, 40, 40	90, 255, 255
Небо	210, 40, 40	260, 255, 255
Пісок на пляжі	20, 30, 70	40, 70, 90
Сніг	0, 0, 200	30, 40, 255
Гірські вершини	20, 40, 40	40, 255, 255

На підставі вищезазначеної інформації буде розроблено аналізатор для розпізнавання об'єктів. Він буде визначати зазначені об'єкти на зображенні, перетворюючи кожен піксель у відповідний колірний простір. Після цього, використовуючи діапазон прийнятних кольорів, аналізатор буде визначати, чи присутній описаний об'єкт на конкретному пікселі чи ні.

Недоліки цього методу будуть схожі на ті, які виникають при визначенні людської шкіри на зображенні. Багато об'єктів у реальному світі можуть мати кольори, схожі на кольори рослин. Найпростіший спосіб вирішення цієї проблеми полягає в явному визначенні меж об'єктів на зображенні.

Після визначення об'єктів аналізатор може застосовувати додаткові методи, такі як фільтрація шуму чи використання алгоритмів машинного навчання для поліпшення точності визначення об'єктів.

Наступні кроки алгоритму будуть аналогічні до тих, які використовуються при реалізації базового алгоритму дискретного вейвлет-перетворення. Вони включають в себе подальші етапи обробки та аналізу зображення з метою вдосконалення визначення об'єктів та зменшення можливих помилок.

На Рисунок 2.2 зображено блок-схему процесу вбудовування секретних даних використовуючи покращений метод приховування інформації.

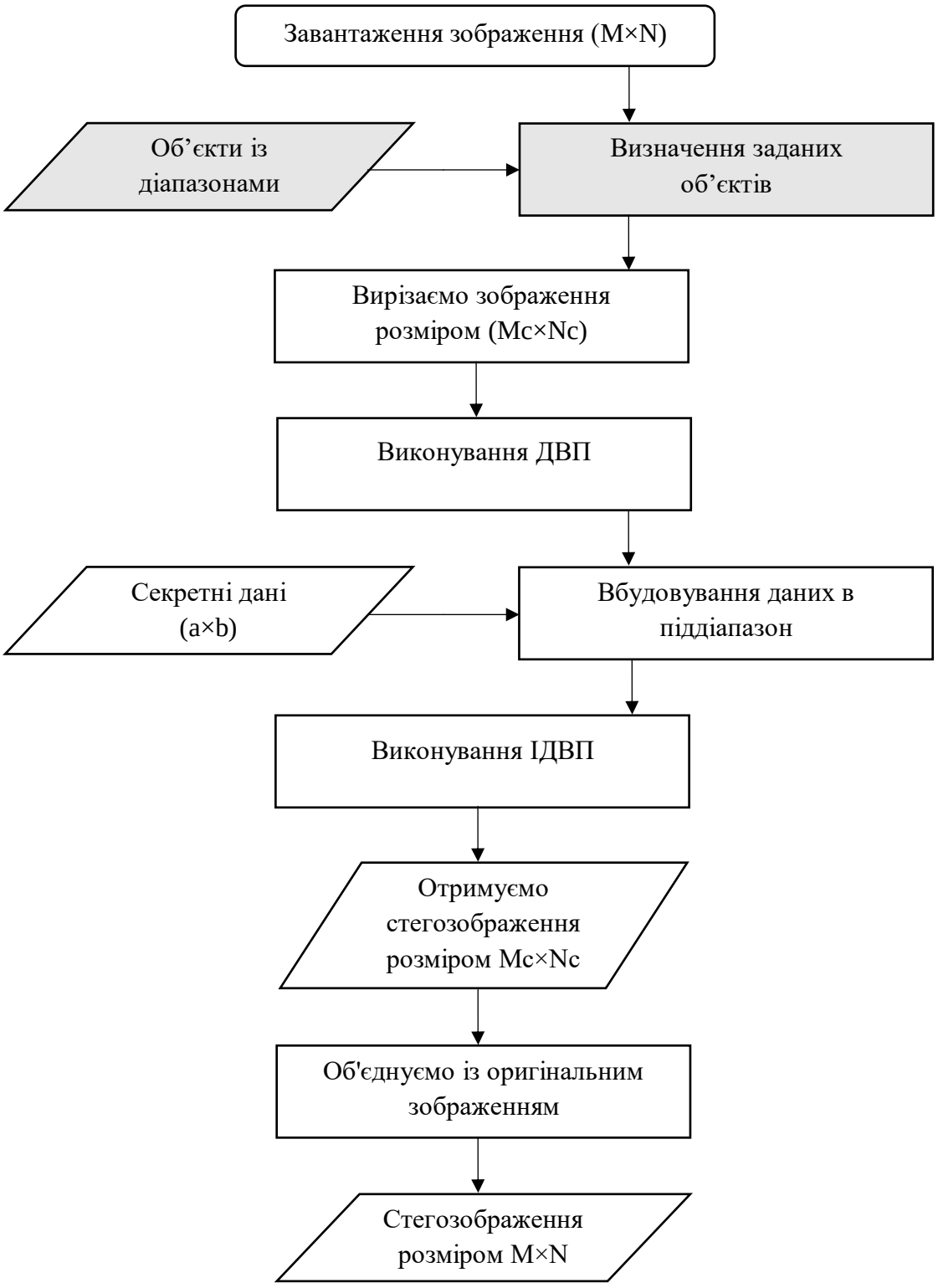


Рисунок 2.2. Блок-схема покращеного процесу вбудовування даних

Базуючись на вище зазначені інформацію в було розроблено алгоритм та покращено методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення за допомогою.

2.4 Висновки до розділу 2

В даному розділі було проаналізовано можливі напрямки вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

Було розроблено алгоритм вдосконалення стеганографічного методу та проаналізовано критерії за допомогою яких буде відбуватися визначення об’єктів на зображенні. Також було наведено таблицю з діапазонами кольорів для різних об'єктів і розроблено алгоритм аналізатору для їх розпізнавання на зображеннях. Було покроково описано як саме має працювати вдосконалений алгоритм.

Виходячи із поставлених завдань роботи, для реалізації програмного додатку було обрані програмні засоби: мова програмування C#; середовище програмування Visual Studio 2022; інтерфейс програмування додатків Windows Forms; програмна технологія для створення додатків .NET Framework. Детальний покроковий опис практичної реалізації розробки опишемо у наступному розділі.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО АЛГОРИТМУ ВДОСКОНАЛЕННЯ СТЕГАНОГРАФІЧНОГО МЕТОДУ ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЦИФРОВИХ ЗОБРАЖЕННЯХ

В даному розділі опишемо етапи практичної реалізації програмної розробки, що призначена для підвищення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення. В ході написання розділу буде здійснено розробку графічного користувацького інтерфейсу, описано особливості програмної реалізації додатку на основі обраних засобів програмування, таких як C#, Visual Studio 2022, Windows Forms та .NET Framework; наведено інструкцію користувача для роботи з програмою, а також проведено тестування вдосконаленого алгоритму на основі програмної розробки для визначення практичних результатів роботи [33].

3.1 Обґрунтування вибору мови та середовища реалізації запропонованого вдосконаленого методу

У наш час існує різноманіття мов програмування, деякі з них широко використовуються, тоді як інші застосовуються в обмежених областях. Ці мови класифікують за рівнем абстракції на високорівневі та низькорівневі. Мови високого рівня працюють з концепціями, що ближчі до людського розуміння, такими як об'єкти, змінні та функції. Низькорівневі мови оперують концепціями, близькими до машини, такими як байти, адреси та інструкції. Зазвичай, текст програм на високорівневих мовах набагато коротший, ніж у випадку такої самої програми на низькорівневій мові, але програма може мати більший обсяг. Розробка програм на високорівневих мовах є простішою, менше можливостей для помилок, і це допомагає скоротити час розробки. До провідних мов програмування відносяться C, C++, Delphi, C# та Java [34].

Для створення додатка було обрано мову програмування C#, оскільки вона є простою, багатофункціональною та набуває широкої популярності. У

ній поєднано переваги різних мов, а швидкодія виконання наближається до мови Assembler. Завдяки зручному об'єктно-орієнтованому дизайну C# є ідеальним вибором для швидкої розробки різноманітних компонентів, від високорівневої бізнес-логіки до системних додатків, що використовують низькорівневий код. Також слід відзначити, що C# орієнтований на Web: використовуючи вбудовані конструкції мови, компоненти можуть легко перетворюватися на Web-сервіси, доступні з Інтернету за допомогою будь-якої мови та операційної системи. Використання передових Web-технологій, таких як XML (Extensible Markup Language) і SOAP (Simple Object Access Protocol), додає до C# додаткові можливості і переваги. Середовище розробки Web-сервісів дозволяє програмістам розглядати існуючі Web-додатки як на рідні об'єкти C#, сприяючи легкій інтеграції Web-сервісів у програми [35].

Дуже часто спостерігається взаємозв'язок: чим більше мова захищена і стійка до помилок, тим менше продуктивність програм, написаних на ній. Для ілюстрації можна порівняти Assembler і Java. В першому випадку можна досягти фантастичної швидкості програми, але виникне потреба в довготривалому налаштуванні для правильної роботи на різних комп'ютерах. У випадку Java, навпаки, досягається висока захищеність і незалежність від платформи, але швидкість програми може не відповідати уявленням про швидкість, яку може надати окремий клієнтський додаток.

У C#, як сучасній мові, є характерні особливості для уникнення можливих помилок. Наприклад, всі змінні автоматично ініціалізуються середовищем і мають типову захищеність, що дозволяє уникнути невизначених ситуацій, якщо програміст забуде ініціалізувати змінну в об'єкті або спробує здійснити неприпустиме перетворення типів. Також в C# вжито заходів для уникнення помилок при оновленні програмного забезпечення. Зміна коду може непередбачувано змінити суть самої програми, і для подолання цієї проблеми C# включає підтримку сумісності версій. На відміну від C++ і Java, якщо метод класу змінено, це повинно бути явно вказано. Це дозволяє уникнути помилок у коді і забезпечити гнучку сумісність версій.

Також варто відзначити нововведення, такі як native підтримка інтерфейсів і спадкоємство інтерфейсів, які дозволяють розробляти складні системи та розширювати їх з часом [36].

В C# введена уніфікована система типів, що дозволяє розглядати кожен тип як об'єкт. Незалежно від того, чи це клас, структура, масив або вбудований тип, можна звертатися до нього як до об'єкта. Об'єкти організовані в простори імен, що дозволяє програмно звертатися до них. Замість переліку файлів, що включаються, в заголовках програми потрібно вказати, які простори імен ви хочете використовувати для доступу до об'єктів і класів всередині них. Вираз using в C# дозволяє не вказувати кожен раз назву простору імен при використанні класу з нього.

Сучасні можливості C# проявляються і в нових підходах до полегшення процесу відлагодження програми. У C++ традиційним засобом відлагодження є розмітка обширних частин коду директивами #ifdef тощо. У C#, використовуючи атрибути, орієнтовані на умовні слова, можна швидше писати код, придатний для відлагодження.

Класи є основним засобом організації даних в C#. Будь-яка програма, написана на цій мові, повинна бути класом, що робить C# "справжньою" об'єктно-орієнтованою мовою, на відміну від, наприклад, C++, де використання об'єктів можливе, але необов'язкове.

Одним з важливих принципів об'єктно-орієнтованого підходу є інкапсуляція даних, яка передбачає, що внутрішній устрій класу і конкретна реалізація його методів повинні бути невідомі зовнішнім споживачам. C# має розвинені засоби підтримки інкапсуляції за допомогою обширного набору модифікаторів доступу, таких як public, protected, private, internal.

У C# конструктори використовуються при створенні конкретних екземплярів класу, зазвичай для ініціалізації значень, що використовуються при подальшій роботі з класом. Конструктори не повертають значень. Якщо клас не має явного опису конструктора, компілятор автоматично генерує порожній конструктор, який викликає базовий клас (якщо він існує).

Важливою відмінністю C# від C++ є відсутність деструкторів у звичайному розумінні. Звільнення пам'яті в .NET відбувається за допомогою механізму зборки сміття, тому явне знищення об'єктів в C# не передбачено. Замість цього використовуються завершувачі (finalizers), які викликаються системою збирання сміття перед знищенням об'єкта. Також, для надійності, в C# рекомендується виділяти завершуючі дії в окремий метод з іменем Close або Dispose.

Усі активні операції програм на C# виконуються в методах класів. Ці методи можуть приймати параметри та повертати значення. При передачі параметрів в C# треба явно вказати спосіб передачі – за значенням або за посиланням; у випадку передачі за посиланням, змінній повинно передувати ключове слово ref. Крім того, в C# передбачена можливість повертати більше одного значення з методу – для цього окрім явного повернення значення методу, треба описати один або декілька параметрів методу з ключовим словом out. Компілятор C# перевіряє, що ref-параметри ініціалізовані перед викликом методу, а також що out-параметри отримують значення до виходу з методу [37].

З точки зору перевантаження методів, важливою особливістю C# є те, що методи за замовчуванням не є віртуальними. Це зроблено для уникнення помилок, пов'язаних з випадковим перевизначенням успадкованих функцій. Крім того, в C# є два способи перевизначення віртуального методу: використовуючи ключове слово override базовий метод стає недоступним, а використовуючи ключове слово new базовий метод все ще можна викликати шляхом явного приведення до типу базового класу.

У об'єктно-орієнтованому програмуванні доступ до даних можна організовувати через методи доступу get і set. До недавнього часу ця рекомендація не підтримувалася мовами програмування, але в C# така можливість нарешті з'явилася. Тепер звичайний опис поля можна доповнити методами доступу get і set, і тоді при будь-якому читанні або привласненні значення полю обов'язково виконуватиметься функціональність, записана в

цих методах доступу. Методи доступу зручні в тих випадках, коли необхідно перевірити допустимість привласненого значення або наявність прав доступу запрошуючого застосування до даного поля [38].

Перевантаження операторів є однією з суперечливих можливостей у сучасних мовах. Деякі програмісти вважають, що перевантаження операторів призводить до помилок. Інші вважають, що це корисний механізм для поліпшення читаності коду. Незалежно від цього, перевантаження операторів стало невід'ємною частиною C#. Більшість класів в C# за замовчуванням перевантажують оператор порівняння (операція ==, яка практично завжди викликає метод Equals успадкований від System.Object).

Отже, обрання C# для розробки додатку обумовлено його потужними можливостями, швидкістю розробки та зручністю в плані обслуговування та підтримки проекту.

3.2 Програмна реалізація додатку на основі вдосконаленого методу

В даному розділі опишемо основні фрагменти коду, що були реалізовані в роботі з метою розробки програмного додатку для вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

HSV – це кольорова модель, яка використовується для представлення кольорів у формі, зручній для людського сприйняття. HSV широко використовується в графічних редакторах та програмах обробки зображень, оскільки вона забезпечує зручний спосіб вираження та розуміння кольору, що важливо в багатьох візуальних додатках. Для конвертації зображення в HSV розроблено дві функції ConvertToHSV та RGBtoHSV.

```
static Bitmap ConvertToHSV(Bitmap originalImage)
{
   int width = originalImage.Width;
   int height = originalImage.Height;
   Bitmap hsvImage = new Bitmap(width, height);
   for (int i = 0; i < width; i++)
```

```
    {   for (int j = 0; j < height; j++)   {
            Color originalColor = originalImage.GetPixel(i, j);
            Color hsvColor = RGBtoHSV(originalColor);
            hsvImage.SetPixel(i, j, hsvColor);
    }   }   return hsvImage;
}
static Color RGBtoHSV(Color rgbColor)
{
    float r = rgbColor.R / 255.0f;
    float g = rgbColor.G / 255.0f;
    float b = rgbColor.B / 255.0f;
    float max = Math.Max(r, Math.Max(g, b));
    float min = Math.Min(r, Math.Min(g, b));
    float h, s, v;
    if (max == min){ h = 0; }
    else if (max == r)
    { h = (60 * (g - b) / (max - min) + 360) % 360; }
    else if (max == g)
    { h = (60 * (b - r) / (max - min) + 120); }
    else
    { h = (60 * (r - g) / (max - min) + 240); }
    if (max == 0) { s = 0; }   else { s = (max - min) / max; }
    v = max;   h = (h / 360) * 255;
    return Color.FromArgb((int)h, (int)(s * 255), (int)(v * 255));
}
```

Також в додатку були встановлені константи значень для визначення меж спектру кольорів для визначення об’єктів на зображенні.

```
var objects = new[]
{
    new { Name = "Рослини", HueMin = 30, HueMax = 90,
      SaturationMin = 40, SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
    new { Name = "Небо", HueMin = 210, HueMax = 260, SaturationMin = 40,
      SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
    new { Name = "Пісок на пляжі", HueMin = 20, HueMax = 40, SaturationMin = 40,
```

```
        SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
    new { Name = "Сніг", HueMin = 0, HueMax = 30, SaturationMin = 0,
        SaturationMax = 40, ValueMin = 200, ValueMax = 255 },
    new { Name = "Гірські вершини", HueMin = 20, HueMax = 40, SaturationMin = 40,
        SaturationMax = 255, ValueMin = 40, ValueMax = 255 }
};
```

За визначення зелених областей на зображенні відповідають функції DetectObjectCoordinates та IsPixelInObjectRange.

```
static List<Tuple<int, int>> DetectObjectCoordinates(Bitmap hsvImage, object obj)
{
    List<Tuple<int, int>> coordinates = new List<Tuple<int, int>>();
    for (int x = 0; x < hsvImage.Width; x++)
    {
        for (int y = 0; y < hsvImage.Height; y++)
        {
            Color hsvColor = hsvImage.GetPixel(x, y);
            int pixelHue = (int)(hsvColor.GetHue() * 2);
            int pixelSaturation = (int)(hsvColor.GetSaturation() * 255);
            int pixelValue = (int)(hsvColor.GetBrightness() * 255);
            if (IsPixelInObjectRange(pixelHue, pixelSaturation, pixelValue, obj))
            { coordinates.Add(new Tuple<int, int>(x, y)); }
        }
    }
    return coordinates;
}
static bool IsPixelInObjectRange(int hue, int saturation, int value, object obj)
{
    return hue >= obj.HueMin && hue <= obj.HueMax &&
        saturation >= obj.SaturationMin && saturation <= obj.SaturationMax &&
        value >= obj.ValueMin && value <= obj.ValueMax;
}
```

Основна ідея використання ДВП в стеганографії полягає в тому, що вейвлет-коефіцієнти містять інформацію про деталі та структуру зображення. Додавання або модифікація коефіцієнтів у вейвлет-домені може призвести до

непоміченої зміни в оригінальному зображенні. Функціонал дискретного вейвлет-перетворення було винесено в дві функції HaarWaveletTransform2D та HaarWaveletTransform.

```
static void HaarWaveletTransform2D(Bitmap image)
{
   int width = image.Width;
   int height = image.Height;
   for (int i = 0; i < height; i++)
   {   HaarWaveletTransform(image, i, 0, width, 1);   }
   for (int j = 0; j < width; j++)
   {   HaarWaveletTransform(image, 0, j, height, width);   }
}
static void HaarWaveletTransform(Bitmap image, int startRow,
int startCol, int length, int stride)
{
   for (int step = length / 2; step >= 1; step /= 2)
   {
      for (int i = 0; i < step; i++)
      {
         int x = startRow + i * 2;
         for (int j = startCol; j < startCol + stride; j++)
         {
            double sum = image.GetPixel(x, j).GetBrightness()
                        + image.GetPixel(x + 1, j).GetBrightness();
            double diff = image.GetPixel(x, j).GetBrightness()
                        - image.GetPixel(x + 1, j).GetBrightness();
            Color sumColor = ColorFromBrightness(sum);
            Color diffColor = ColorFromBrightness(diff);
            image.SetPixel(x / 2, j, sumColor);
            image.SetPixel(x / 2 + step, j, diffColor);
         }
      }
   }
}
```

Приховування інформації в обрану зону відбувається за допомогою функції EncodeSecretMessage.

```
static void EncodeSecretMessage(Bitmap greenChannel, string message)
{
   byte[] messageBytes = Encoding.UTF8.GetBytes(message);
   int width = greenChannel.Width;
   int height = greenChannel.Height;
   int messageIndex = 0;
```

```
    for (int i = 0; i < width; i++)
    {
        for (int j = 0; j < height; j++)
        {
            Color pixelColor = greenChannel.GetPixel(i, j);
            int greenValue = pixelColor.G;
            if (messageIndex < messageBytes.Length)
            {
                greenValue = (greenValue & 0xFE) |
                                ((messageBytes[messageIndex] >> 7) & 0x01);
                messageIndex++;
            }
            greenChannel.SetPixel(i, j, Color.FromArgb(greenValue, greenValue, greenValue));
        }
    }
}
```

Інверсне дискретне вейвлет-перетворення в стеганографії може використовуватися для витягання прихованої інформації з зображень, які були попередньо модифіковані за допомогою ДВП. Основна ідея полягає в тому, що прихована інформація впроваджується у вейвлет-домені за допомогою ДВП, а потім може бути вилучена за допомогою ІДВП. Важливо враховувати, що при використанні ДВП та ІДВП в стеганографії, слід уникати великих змін у вейвлет-коефіцієнтах, щоб уникнути помітних артефактів та забезпечити надійність. Для інверсного дискретне вейвлет-перетворення визивається функція InverseHaarWaveletTransform2D та InverseHaarWaveletTransform.

```
static void InverseHaarWaveletTransform2D(Bitmap image)
{
    int width = image.Width; int height = image.Height;
    for (int j = 0; j < width; j++)
    { InverseHaarWaveletTransform(image, 0, j, height, width); }
    for (int i = 0; i < height; i++)
    { InverseHaarWaveletTransform(image, i, 0, width, 1); }
}
```

```
static void InverseHaarWaveletTransform(Bitmap image, int startRow,
   int startCol, int length, int stride)
{
   for (int step = 1; step <= length / 2; step *= 2)
   {
      for (int i = 0; i < step; i++)
      {
         int x = startRow + i * 2;
         for (int j = startCol; j < startCol + stride; j++)
        {
            Color sumColor = image.GetPixel(x / 2, j);
            Color diffColor = image.GetPixel(x / 2 + step, j);
            double sum = sumColor.GetBrightness();
            double diff = diffColor.GetBrightness();
            double avg = (sum + diff) / 2;
            double delta = (sum - diff) / 2;
            image.SetPixel(x, j, ColorFromBrightness(avg + delta));
            image.SetPixel(x + 1, j, ColorFromBrightness(avg - delta));
         }
} } }
```

Фінальною операцією відповідає за інтегрування опрацьованого зображення з прихованим повідомленням в оригінальне зображення. Для цього визивається функція ReplaceGreenChannel. Вона повертає координати вмонтованого зображення.

```
static Rectangle ReplaceGreenChannel(Bitmap hsvImage, Bitmap newGreenChannel)
{
   int width = hsvImage.Width;   int height = hsvImage.Height;
   int replacedX = width;   int replacedY = height;
   int replacedWidth = 0;   int replacedHeight = 0;
   for (int i = 0; i < width; i++)
   {
      for (int j = 0; j < height; j++)
      {
         Color originalColor = hsvImage.GetPixel(i, j);
```

```
            int newGreenValue = newGreenChannel.GetPixel(i, j).G;
            Color newColor =
                        Color.FromArgb(originalColor.R, newGreenValue, originalColor.B);
            hsvImage.SetPixel(i, j, newColor);
            if (newGreenValue != originalColor.G)
            {
               replacedX = Math.Min(replacedX, i);
               replacedY = Math.Min(replacedY, j);
               replacedWidth = Math.Max(replacedWidth, i - replacedX + 1);
               replacedHeight = Math.Max(replacedHeight, j - replacedY + 1);
      } } }
      return new Rectangle(replacedX, replacedY, replacedWidth, replacedHeight);
   }
```

Для зчитування прихованої інформації потрібно завантажити зображення і вказати вхідні значення де було приховано повідомлення. Для цього використовується функція DecodeSecretMessage.

```
   static string DecodeSecretMessage(Bitmap greenChannel, Rectangle regionToDecode)
   {
      int startX = regionToDecode.X;   int startY = regionToDecode.Y;
      int endX = startX + regionToDecode.Width;
      int endY = startY + regionToDecode.Height;
      StringBuilder decodedMessage = new StringBuilder();
      for (int i = startX; i < endX; i++)
      {
         for (int j = startY; j < endY; j++)
         {
            Color pixelColor = greenChannel.GetPixel(i, j);
            int greenValue = pixelColor.G;
            decodedMessage.Append((byte)(greenValue & 0x01));
      } }
      string decodedString = Encoding.UTF8.GetString(ConvertBinaryStringToBytes(
         decodedMessage.ToString()));
      return decodedString;
   }
```

3.3 Розробка графічного інтерфейсу програмної розробки

При розробці будь-якого програмного додатку важливо звертати увагу на користувацький інтерфейс (UI) – це візуальна частина розробки, яка визначає спосіб взаємодії користувача з програмним додатком, пристроєм чи системою. Користувацький інтерфейс включає в себе різноманітні елементи, такі як кнопки, поля введення, меню, панелі, вікна та інші, які дозволяють користувачеві взаємодіяти з програмою та керувати її функціоналом [39].

Основними завданнями користувацького інтерфейсу є полегшення виконання основних функцій додатку, навігація по ньому, представлення необхідної інформації та забезпечення взаємодії між програмним забезпеченням та користувачем.

З урахуванням цих особливостей ключовими вимогами до розробки користувацького інтерфейсу є

- Простота та легкість у використанні: користувачеві повинно бути зрозуміло, як користуватися по програмою та взаємодіяти з її елементами.
- Зрозумілість дизайну: інтерфейс повинен бути логічним, простим та інтуїтивно зрозумілим для користувача; елементи управління повинні мати зрозумілі та доступні назви та символи.
- Ефективність та продуктивність: інтерфейс повинен бути скомпонований лаконічно та зрозуміло, зменшуючи кількість кроків та розташовуючи необхідні елементи управління в логічній послідовності.

Вимоги та особливості розробки інтерфейсу варіюють залежно від потреб та призначення програмного додатку. Оскільки розроблюваний додаток у цій роботі призначений для приховування інформацію в зображенні та має практичну цінність в прямому використанні, інтерфейс додатку повинен мати стриманий дизайн, зрозумілі функціональні клавіші та лаконічні діалогові форми для взаємодії користувача та програми .

Розглянемо структуру розроблений додаток. Після запуску програми, відкривається її головне вікно. Структурно додаток можна розділити на дві частини:

- Encode – вікно котре відповідає за збір вхідних даних та шифрування секретної інформації (Рисунок 3.1). Після успішного шифрування та збереження зображення, відображається повідомлення із даними котрі потрібні для вдалого розшифрування (Рисунок 3.2).
- Decode – вікно котре відповідає за збір даних та розшифрування секретної інформації (Рисунок 3.3). Після успішного розшифрування, відображається повідомлення з прихованим повідомленням (Рисунок 3.4).
- Ця структура передбачає взаємодію користувача з двома окремими функціональними частинами додатка: кодуванням та декодуванням. Важливо врахувати, що в описаних функціональних блоках можуть бути ще додаткові елементи, такі як кнопки, поля введення, повідомлення про помилки і т. д., які сприяють зручності використання додатка.

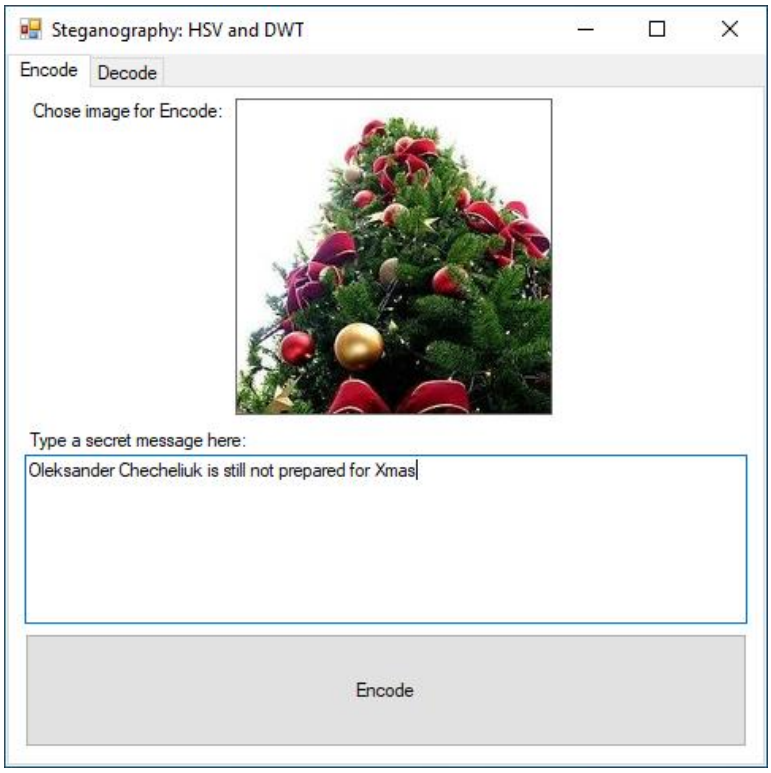


Рисунок 3.1 Зображення Encode вікна

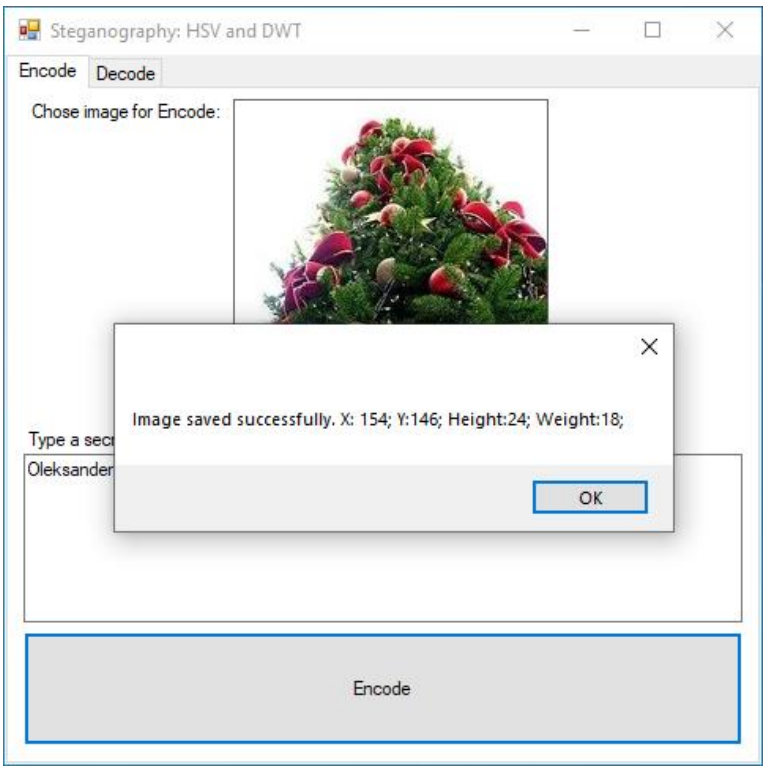


Рисунок 3.2 Зображення повідомлення із успішним шифруванням

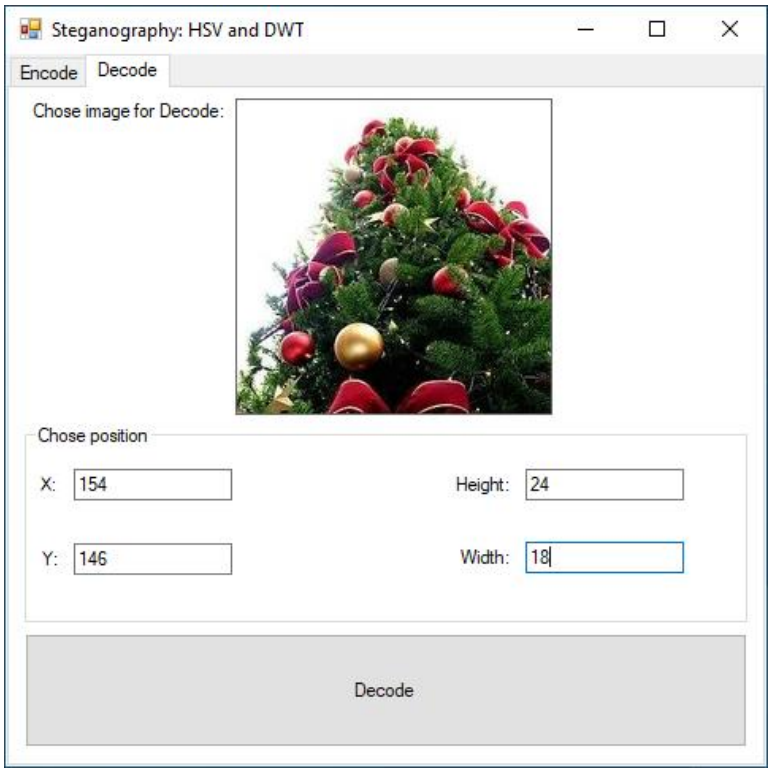


Рисунок 3.3 Зображення Decode вікна

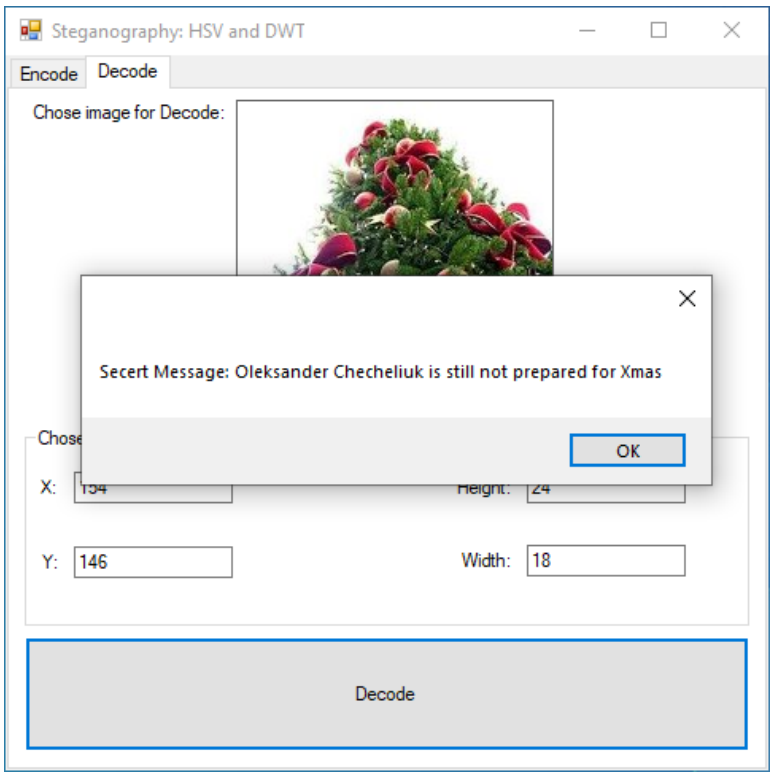


Рисунок 3.4 Зображення розшифрованого повідомлення

Таким чином, у даному підрозділі на прикладі практичного застосування було описано інструкцію для роботи користувача з додатком, основна функція якого полягає у практичній реалізації вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

3.4 Аналіз стійкості вдосконаленого методу

У цьому підрозділі ми аналізуємо PSNR, показник якості та здатність до оцінки ефективності в запропонованій схемі. У схемі приховування даних PSNR вимірюється ступінь спотворення між обкладинкою та зображенням стего, використовуючи таке рівняння [14]:

$$PSNR = 10 * log_{10} \frac{255^2}{MSE}. \quad (3.1)$$

Середньоквадратична помилка обчислюється за наступним рівнянням:

$$MSE = \sum_{i=1}^{W*H} \frac{(p_i - p_i`)^2}{W*H}. \quad (3.2)$$

Якщо значення PSNR більше 30 дБ, спотворення зображення неможливо виявити людськими очима. Індекс якості є показником кореляції між двома зображеннями. Якщо показник якості дорівнює 1, два зображення однакові. І навпаки, якщо показник якості дорівнює -1, то два зображення – різні. Індекс якості показаний у наступному рівнянні:

$$Q = \frac{4\delta_{xy}\overline{p_x^2}\overline{p_x^2}}{(\delta_x^2+\delta_y^2)(\overline{p_x^2}+\overline{p_y^2})}. \quad (3.3)$$

Рівняння для кожного елемента рівняння такі:

$$\overline{p_x} = \frac{1}{wh}\sum_{i=0}^{wh-1} p_i, \quad (3.4)$$

$$\overline{p_y} = \frac{1}{wh}\sum_{i=0}^{wh-1} p`_i, \quad (3.5)$$

$$\delta_x^2 = \frac{1}{wh-1}\sum_{i=0}^{wh-1}(p_i - \overline{p_x})^2, \quad (3.6)$$

$$\delta_y^2 = \frac{1}{wh-1}\sum_{i=0}^{wh-1}(p`_i - \overline{p_y})^2, \quad (3.7)$$

$$\delta_{xy} = \frac{1}{wh-1}\sum_{i=0}^{wh-1}(p_i - \overline{p_x}) - (p`_i - \overline{p_y}) \quad (3.8)$$

Індекс якості визначається як поєднання втрат кореляції, спотворення яскравості та контрастності, що перевизначається як таке рівняння:

$$Q = \frac{\delta_{xy}}{\delta_x\delta_y} * \frac{2\overline{p_x^2}\overline{p_y^2}}{\overline{p_x^2}+\overline{p_y^2}} * \frac{2\delta_x\delta_y}{\delta_x^2+\delta_y^2}. \quad (3.9)$$

Коефіцієнт кореляції між двома зображеннями становить $\frac{\delta_{xy}}{\delta_x\delta_y}$. Яскравість між двома зображеннями вимірюється за допомогою $\frac{2\overline{p_x^2}\overline{p_y^2}}{\overline{p_x^2}+\overline{p_y^2}}$, а подібність двох зображень вимірюється за допомогою $\frac{2\delta_x\delta_y}{\delta_x^2+\delta_y^2}$. Ємність вбудовування означає розмір секретних даних, які можуть бути вбудовані в зображення обкладинки. На рисунку 3.5 показано кольорові зображення розміром 512×512, використані в експерименті.

Рисунок 3.5 Зображення, використані в експерименті

Таблиця 3.1 порівнює PSNR вдосконаленого методу з базовим методом за формулою 2.12. Базовий метод використовує зображення дитини для вбудовування секретних даних, тоді як вдосконалений метод використовує зображення рослини.

Таблиця 3.1 Порівняння значень методів

Назва методу	Зображення	PSNR (дБ)	MSE	Ємність (біт)
Базовий	Дитина	69.5104	0.00727844	1248890
Вдосконалений	Рослина	70.3077	0.00605776	1225240
Вдосконалений	Небо	68.3502	0.00734539	1238770
Вдосконалений	Пісок на пляжі	70.1243	0.00648375	1215370
Вдосконалений	Сніг	69.4212	0.00675345	1256870
Вдосконалений	Гірські вершини	69.8043	0.00743641	1268763

Отже, результати проведених експериментів показали, що такі показники як PSNR, MSE та ємність у вдосконаленому методі, в порівнянні із базовим методами, дали схожі результати із невеликою відмінністю. З цього випливає те що наш покращений метод працює аналогічно як базовий проте він приймає більший діапазон зображення. Дане покращення експоненціально збільшить області вбудовування секретних даних.

3.5 Висновки до розділу 3

Таким чином, у даному розділі на основі обраних засобів програмування таких як C#, Visual Studio 2022, Windows Forms та .NET Framework було здійснено програмну реалізацію додатку для підвищення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

В ході написання розділу було здійснено розробку графічного користувацького інтерфейсу програмного додатку із врахуванням особливостей розробки; наведено основні фрагменти коду, що були написані для реалізації основного функціоналу розробки.

Показано користувацький інтерфейс, представлено інструкцію користувача, яка полегшить роботу користувачів програми. Також покроково описано роботу програми з призначенням кожного елементу інтерфейсу.

Експериментальні результати продемонстрували, що запропонована схема має високу здатність до вбудовування та прийнятну непомітність у якості візуального зображення. Базуючись на цьому було запропоновано використовувати комбінований метод котрий експоненціально збільшить області вбудовування секретних даних.

4 ЕКОНОМІЧНА ЧАСТИНА

Метою даного розділу є проведення дослідження економічного потенціалу розробки, зокрема, оцінювання комерційного потенціалу; прогнозування витрат на виконання наукової роботи та впровадження її результатів; прогнозування комерційних ефектів від реалізації результатів розробки та розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Результати даного дослідження уможливлюють прийняття рішення про економічну доцільність розробки програмного засобу з використанням вдосконаленого методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

4.1 Оцінювання комерційного потенціалу програмного забезпечення

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Результатом магістерської кваліфікаційної роботи є розробка програмного засобу на основі вдосконаленого методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення, що практично реалізований у вигляді програмного додатку.

Для проведення технологічного аудиту залучено трьох незалежних експертів.

Оцінювання комерційного потенціалу буде здійснено за критеріями, що наведені в таблиці 4.1 [40].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження табл. 4.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виро-
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на	Необхідно отримання великої кількості документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	2	3
Наявність аналогів на ринку	3	2	2
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	2	2	2
Експлуатаційні витрати	3	3	3
Ринок збуту	2	2	3
Конкурентоспроможність	3	2	3
Фахівці з технічної і комерційної реалізації	3	3	4
Фінансування	4	4	3
Матеріально-технічна база	2	4	4
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	3
Сума	35	35	37
Середньоарифметична сума балів	(35+35+37) / 3 = 36		

За даними таблиці 4.2 можна зробити висновок, що рівень комерційного потенціалу даної розробки становить 36та відповідно до таблиці 4.3 є «вищим середнього».

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Потенційними споживачами програмного забезпечення на основі представленого методу можуть бути два види користувачів: приватний та корпоративний сектор. Запропоноване програмне забезпечення може зацікавити приватних користувачів, адже захистити свої авторські права на зображення хоче кожен, чиє зображення є його цінною власністю та щоб при

цьому, для якості самого зображення, була нанесена мінімальна шкода. Так само і керівники різних фірм чи установ, придбавши дане ПЗ, зможуть забезпечити собі захист авторського права на зображення, які допомагають їм вести свій бізнес.

Бажаним каналом реалізації розробки є прямий канал, що являє собою переміщення товару від виробника до споживача без залучення інших сторін. Даний канал реалізації було обрано у початковим етапом наукової розробки і не великими обсягами продаж. Таким чином вдасться мінімізувати витрати на проміжні ланки продажу.

Новизна розробки у порівнянні з аналогами становить вдосконаленого методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення. Це , в свою чергу, позитивно впливає на ціну і відповідно на рівень конкуренції.

Додатковим є можливість, у випадку зацікавленості покупця, подальшого вдосконалення запропонованого методу.

Позитивним фактом також є відсутня необхідність у залученні значних джерел фінансування.

Врахувавши цінову політику на ринку, а також враховуючи той факт, що розробка не вимагає багато затрат, ціна розробки буде значно нижче існуючих аналогів, що позитивно вплине на продажі розробки. На першому етапі доцільно буде встановити ціну, яка дозволить отримувати мінімальну дохідність, в майбутньому ціну можна підвищити до оптимальної.

4.2 Прогнозування витрат на виконання наукової роботи

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, за такими статтями та формулами, приймаючи до уваги те, що для розробки інформаційної технології було залучено одного розробника програмного забезпечення.

Розрахунок основної заробітна плата розробників, яка розраховується за формулою:

$$З_р = \sum_{1}^{n} t_i \cdot C_i \qquad (4.1)$$

де М – місячний посадовий оклад конкретного розробника грн.;

$Т_р$ – число робочих днів в місяці, 23 днів;

t – число днів роботи розробник.

Результати розрахунків зведемо до таблиці 4.4.

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	70000	3043,47	23	70000
Розробник	50000	2173.91	23	50000
Всього				120000

Витрати на основну заробітну плату робітників ($З_р$) розраховуються на основі норм часу, які необхідні для виконання даної роботи, розраховуються за формулою:

$$З_р = \sum_{1}^{n} t_i \cdot C_i \quad , \qquad (4.2)$$

де t_i – норма часу (трудомісткість) на виконання конкретної роботи, годин;

n – число робіт по видах та розрядах;

Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує відповідну роботу, грн./год.

C_i визначається за формулою:

$$C_i = \frac{М_м \cdot К_i \cdot К_с}{Т_р \cdot t_{зм}}, \qquad (4.3)$$

де $М_н$ – мінімальна місячна оплата праці, грн., $М_н$ = 6700 грн. (листопад 2023 року);

$К_і$ – тарифний коефіцієнт робітника відповідного розряду;

Кс – коефіцієнт співвідношень, який установлений в даний час Генеральною тарифною угодою між Урядом України і профспілками;

$Т_р$ – число робочих днів в місяці, $Т_р$ = 23 дні;

$Т_{зм}$ – тривалість зміни, $Т_{зм}$ = 8 годин.

$$С_і,=\frac{6700\times2,4\times1,8}{23\times8}=\frac{28944}{184}=157,3 \text{ (грн./год)}$$

Погодинна тарифна ставка відповідно до формули 4.3 становить 157,3 грн./год. Розраховані витрати на заробітну плату відображено в табл. 4.5.

Таблиця 4.5 – Витрати на основну заробітну плату робітників

Найменування робіт	Трудомісткість, год.	Розряд роботи	Погодинна тарифна ставка, грн.	Величина оплати на робітника грн.
Організація робочого місця	8	8	157,3	1258,4
Всього	1258,4,			

Додаткова заробітна плату будемо розраховувати як 12 % від основної заробітної плати розробників та робітників відповідно до формули:

$$З_Д = \frac{(З_О+З_Р)\cdot 12\%}{100\%}, \quad (4.4)$$

$З_д$ = (120000 + 1258,4) · 12 % / 100 % = 14551 (грн.)

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$З_н = \frac{(З_О+ З_Р+ З_Д)\cdot 22\%}{100\%}, \quad (4.5)$$

Підставивши отримані значення в попередніх крокам отримаємо наступну суму:

$З_н$ = (120000 + 1258,4 + 14551) · 0.22 = 29878,06 (грн)

При розробці були використані матеріали, кількість та вартість яких зведені в таблицю 4.6.

Таблиця 4.6 – Матеріали, використані для реалізації проєкту

Найменування матеріалу, марка, тип, сорт	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Папір А4	200	3	600
Флешка USB	200	1	200
Папка для паперів	50	2	100
Файли	20	1	20
Ручка	30	4	120
Всього:			1040

Отже, вартість витрачених матеріалів, М = 1040 грн.

Амортизація обладнання, що використовувалось для розробки розраховується за формулою:

$$А = \frac{Ц}{Т_в} \cdot \frac{T}{12} \text{ (грн.)} \quad (4.6)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років

Перелік використаних ресурсів, обладнання, їх балансова вартість та відповідні значення амортизаційних відрахувань відображено в таблиці 4.7.

Таблиця 4.7 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп’ютер	50000	3	1	1388,88
Меблі	11000	4	1	229,16
Приміщення	550000	20	1	2291,66
ПЗ	20000	2	1	833,33
Всього				4743,03

Витрати на силову електроенергію розраховуються за формулою 4.7:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{впі}}{\eta_i}, \quad (4.7)$$

де $Ц_e$ — вартість 1 кВт-години електроенергії, В = 7.5 грн./кВт (у листопаді 2023 році);

W_{yi} — встановлена потужність обладнання, кВт. П = 0.8 кВт;

t — фактична кількість годин роботи обладнання, годин – 184;

$K_{впі}$ — коефіцієнт, що враховує використання потужності, $K_{впі} < 1$;

n_i — коефіцієнт корисної дії обладнання.

Перелік розрахунків відображений в таблиці 4.8

Таблиця 4.8 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Комп'ютер	0,8	184	827,98
Всього			827,98

Витрати які відносяться до статей «Службові відрядження», «Витрати на роботи, які виконують сторонні підприємства, установи і організації», «Інші витрати», «Накладні (загальновиробничі) витрати» вираховуються за формулою яка подібна формулі 4.4. Лише для кожної групи буде встановлено відповідний відсоток.

Для «Службові відрядження» та «Витрати на роботи, які виконують сторонні підприємства, установи і організації» буде встановлено значення 0% так як ці витрати не є актуальні для даного проєкта.

Для «Інші витрати» та «Накладні (загальновиробничі) витрати» буде встановлено значення 100% і результатом буде значення 121258,4 грн. для кожної із статей витрат.

Сума всіх попередніх статей витрат дає загальні витрати на проведення розробки та розраховується за формулою 4.8:

$$B_{заг} = З_o + З_p + З_{дод} + З_н + M + K_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв} \quad (4.8)$$

Отже, загальна сума витрат дорівнює:

$В_{заг}$ = 120000 + 1258,4 + 14551 + 29878,06 + 1040 + 0 + 0 + 0 + 4743,03 + 827,98 + 0 + 0 + 0 + 121258,4 + 121258,4 = 414815,27 грн.

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної МКР здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta} \text{ (грн)} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної МКР $\beta = 0{,}1\ldots0{,}9$. Приймемо β = 0,5, так як розробка, на даний момент часу, знаходиться на стадії дослідного зразка, тоді значення загальних витрат буде дорівнювати:

ЗВ = 414815,27/ 0,5 = 829630,54 грн.

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо збільшення чистого прибутку підприємства $\Delta П_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta П_i = \sum_1^n \left(\Delta Ц_o \cdot N + Ц_o \cdot \Delta N\right)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right), \quad (4.9)$$

де $\Delta Ц_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:

$Ц_о$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт λ = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту, р = 0,25;

υ – ставка податку на прибуток. У 2023 році — 18%.

Моніторинг українського ринку дозволяє зробити висновок, що аналогові програми для захисту ЦЗ представлені, та їх середня вартість зросла протягом попереднього року на 400 грн. Згідно статистики було продано біля 5000 користувацьких копій. Тому при прогнозованій ціні 4000 грн. за користувацьку копій , термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти його ціну на 1000 грн. Кількість реалізованої продукції також збільшиться: протягом першого року — на 1000 копій., протягом другого року — на 800 копій, протягом третього року на 600 копій.

До моменту впровадження результатів наукової розробки, реалізації даного продукту на ринку не було:

$\Delta П_1 = (400 \cdot 5000 + (4000 + 1000) \cdot 1000) \cdot 0.8333 \cdot 0.25 \cdot (1 - 0.18) = 1195785,5$ грн.

$\Delta П_2 = (400 \cdot 5000 + (4000 + 1000) \cdot (1000 + 800)) \cdot 0,8333 \cdot 0,25 \cdot (1 - 0,18) = 1879091,5$ грн.

$\Delta П_3 = (400 \cdot 5000 + (4000 + 1000) \cdot (1000+ 800 + 600)) \cdot 0,8333 \cdot 0,25 \cdot (1 - 0,18) = 2391571$ грн.

Отже, прогнозований комерційний ефект від реалізації результатів розробки за три роки складе 5466448 грн.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Визначимо поточну вартість загального приросту чистого прибутку (ПП) в результаті потенційної інтеграції та комерціалізації науково-технічного розробки, на отримання якого може розраховувати інвестор за формулою (4.10):

$$\text{ПП} = \sum_{1}^{\text{T}} \left(\frac{\Delta \text{П}_\text{i}}{(1+\tau)^t} \right), \tag{4.10}$$

де $\Delta П_i$ – приріст чистого прибутку за кожний із років, у яких наявні результати проведених та виконаних досліджень та розробок, в грн;

T – тривалість періоду, в момент якого стають очевидними результати інтегрованих досліджень та розробок, вимірюється роками.

τ – ставка дисконту, яка враховує річний прогнозований рівень інфляції в країні, $\tau = 0{,}1$;

t – період часу, у роках.

Також, слід зазначити, що зростання прибутку ми будемо отримувати з першого року:

$$\text{ПП} = \left(\frac{1195785{,}5}{(1+0{,}1)^1} \right) + \left(\frac{1879091{,}5}{(1+0{,}1)^2} \right) + \left(\frac{2391571}{(1+0{,}1)^3} \right) = 4436868{,}58 \text{ грн.}$$

Тепер, розрахуємо початкову суму інвестицій (PV), яку потенційний інвестор має спрямувати на реалізацію та комерціалізацію науково-технічної розробки, за допомогою наступної формули (4.11):

$$PV = k_{інв} \cdot ЗВ, \tag{4.11}$$

де $k_{інв}$ – коефіцієнт, що включає витрати інвестора на інтеграцію науково-технічного засобу та його комерціалізацію, враховує такі витрати, як підготовка об’єкта, розробка програми, навчання персоналу, маркетингова діяльність. Як правило, $k_{інв}$ знаходиться в діапазоні від 2 до 5;

$ЗВ$ – загальна сума витрат на здійснення досліджень і розробок та

оформлення їх результатів, грн.

$$PV = 2 \cdot 829630{,}54 = 1659261{,}08 \text{ грн.}$$

За формулою (4.12), визначимо абсолютний економічний ефект, який позначається як E_{abc} або чистий поточний дохід (NPV), який є результатом потенційного впровадження та комерціалізації досліджень і розробок:

$$E_{абс} = ПП - PV, \tag{4.12}$$

$$E_{абс} = 4436868{,}58 - 1659261{,}08 = 2777607{,}5 \text{ грн.}$$

Після інтеграції нашої розробки, $E_{abc} > 0$, це свідчить про те, що впровадження нашого проєкту призведе до позитивного чистого прибутку або економічної вигоди. Це свідчить про те, що наша розробка позитивно вплине на проєкт або підприємство, зробивши його фінансово життєздатнішим і потенційно прибутковішим, ніж це було раніше.

Отже, інвестування коштів у проєкт може бути доцільним.

Для прийняття обґрунтованих рішень щодо інвестицій у дослідження і розробку вкрай важливо оцінити їх відносну (річну) ефективність. Ця оцінка допомагає нам зрозуміти економічну життєздатність цих інвестицій з часом. Відносна ефективність дає зрозуміти річний прибуток або вигоди, які можна очікувати від інвестованого капіталу. Аналізуючи цю відносну ефективність, зацікавлені сторони можуть краще оцінити довгострокові фінансові перспективи ініціатив наукового розвитку та прийняти обґрунтовані інвестиційні рішення. Тому, використовуючи формулу (4.13), розрахуємо відносну (річну) ефективність і порівняємо її з дисконтною ставкою:

$$\mathrm{E_в} = \sqrt[\mathrm{Т_ж}]{1 + \frac{\mathrm{E_{абс}}}{PV}} - 1, \tag{4.13}$$

де $T_ж$ – життєвий цикл наукової розробки, у роках.

$$E_в = \sqrt[3]{1 + \frac{2777607{,}5}{1659261{,}08}} - 1 = 0{,}38.$$

Розрахуємо мінімальну ставку дисконту, за формулою (4.14):

$$\tau_{min} = d + f, \tag{4.14}$$

де d – середньозважена процентна ставка за депозитними операціями в комерційних банках, d = 0,14;

f – ризикованість вкладень, f = 0,05.

$$\tau_{min} = 0,14 + 0,05 = 0,19.$$

Як бачимо, $E_в > \tau_{min}$, а це означає, що інвестор може бути зацікавлений у фінансуванні нашого проєкту.

Тепер, обчислимо термін окупності коштів, інвестованих у наший проєкт, за формулою (4.15):

$$T_{ок} = \frac{1}{\mathrm{E_в}}, \quad (4.15)$$

$$T_{ок} = \frac{1}{0,35} = 2,63 \text{ р.}$$

За нашими розрахунками, $T_{ок}$ < 3-х років, а це вказує на те, що проєкт, швидше за все, генеруватиме позитивні грошові потоки відносно швидко, що робить його вигідною інвестицією з точки зору його здатності окупити початкові витрати протягом короткого періоду часу.

4.5 Висновки до розділу 4

Виконавши дослідження комерційного потенціалу було здійснено оцінювання комерційного потенціалу розробки трьома експертами, відповідно до середньої оцінки було встановлено, що дана розробка має рівень комерційного потенціалу вище середнього.

Було виконано розрахунок прогнозованих загальних витрат на виконання та провадження результатів розробки, що становить 829630,54 грн.

Прогнозований комерційний ефект дозволяє переконатись в доцільності виконаної роботи. Чистий прибуток протягом трьох років від реалізації розробки становить 5466448 грн. Даний прибуток дозволяє окупити затрати на розробку протягом двох років, тобто фінансування даної розробки є доцільним оскільки термін окупності менше трьох років. Додатково дану гіпотезу підтверджує щорічна ефективність вкладених коштів становить 2,63 роки.

ВИСНОВКИ

В даній магістерській кваліфікаційній роботі було проведено вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

В першому розділі було проаналізовано сучасний стан інформації та сфери, які потребують її захисту. Детально розглянуто стеганографічний захист інформації, а саме захист зображення за допомогою ЦВЗ. Розглянуто зображення, як контейнер для вбудовування ЦВЗ. Проаналізовано основні методи захисту інформації, визначено переваги та недоліки кожного з них. Також через ряд перевагу було обрано метод дискретного вейвлет-перетворення.

В другому розділі було проаналізовано можливі напрямки вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

Було розроблено алгоритм вдосконалення стеганографічного методу. В ньому покроково описано як саме має працювати вдосконалений алгоритм.

Були порівняні такі основні показники зображення як PSNR, MSE та ємність для зображень із вбудованим ЦВЗ вдосконаленим методом та його базовою версією.

У третьому розділі на основі обраних засобів програмування таких як C#, Visual Studio 2022, Windows Forms та .NET Framework було здійснено програмну реалізацію додатку. В ході написання розділу було здійснено розробку користувацького інтерфейсу програмного додатку із врахуванням особливостей розробки; наведено основні фрагменти коду, що були написані для реалізації основного функціоналу розробки; описано інструкцію користувача для роботи з програмою.

Експериментальні результати продемонстрували, що запропонована схема має високу здатність до вбудовування та прийнятну непомітність у якості візуального зображення. Базуючись на цьому було запропоновано

використовувати комбінований метод котрий експоненціально збільшить області вбудовування секретних даних.

В четвертому розділі роботи було виконано оцінювання комерційного потенціалу розробки програмного засобу , яке показало, що виконана робота має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

Аналізуючи отримані результати, можна вважати, що в ході виконання роботи досягнута її основна мета, а саме здійснено вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конахович Г. Ф., Пузиренко А. Ю. Комп'ютерна стеганографія теорія і практика – Київ: МК-Прес. -2006.

2. Dumitrescu D., Stan I. M., Simion E. Steganography techniques //IACR Cryptology ePrint Archive. – 2017. – Т. 2017. – С. 341.

3. Romanova A., Toliupa S. Perspective steganographic solutions and their application //Proceedings of the VII Inter University Conference Engineer of XXI Century. – 2017. – Т. 2. – С. 269-278.

4. Yang H., Kot A. C. Data hiding for text document image authentication by connectivity-preserving IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. – IEEE, 2005. – Т. 2. – С. ii/505-ii/508 Vol. 2.

5. Різуненко А. О. Теорія та практика цифрової обробки зображень Полтава: РВВ ПУСКУ. – 2009.

6. Чеховський С. Сучасні методи прихованої передачі інформації через програмне керування випромінюванням комп'ютера. – 2003.

7. Стеганографія. Навчальний посібник. веб-сайт. URL: http://tks.nau.edu.ua/wp-content/uploads/2016/05/Steganografiya.pdf.

8. Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп’ютерної стеганографії : Навчальний посібник для студентів і аспірантів. – Вінниця:ВДТУ, 2003. – 143 с

9. Конахович Г.Ф. Комп’ютерна стеганографія. Теорія і практика / Г.Ф. Конахович, А. Ю. Пузиренко. – Київ: МК-Пресс, 2006. – 288 с.

10. В.Д. Козюра, В.О. Хорошко, М.Є. Шелест, Ю.М. Ткач, О.О. Балюнов Захист інформації в комп’ютерних система – Чернігів 2020

11. Anjali A.Ahejul And U.L.Kulkarni, “A DWT based Approach for Steganography Using Biometrics”,International Conference on Data Storage and Data Engineering, 2010

12. Po-Yueh Chen and Hung-Ju Lin “A DWT Based Approach for Image Steganography”, International Journal of Applied Science and Engineering, 2006. 4, 3: 275-290

13. Parvez MT, Gutub AA. 2011 Vibrant color imagesteganography using channel differences and secretdata distribution. Kuwait J.Sci. Eng

14. Комп’ютерна стеганографія. StudFiles. URL: https://studfile.net/preview/9650047/page:7/

15. Козюра В. Д., Хорошко В. О., Шелест М. Є., Ткач Ю. М., Балюнов О.О. Захист інформації в комп’ютерних системах: підручник. – Ніжин: ФОП Лук’яненко В.В., ТПК «Орхідея», 2020. – 236 с.

16. Бабич І.В., Паламарчук С.А., Паламарчук Н.А., Овсянніков В.В. Огляд стеганографічних методів перетворення інформації в зображеннях // Науково-практичний журнал, „Захист інформації” № 1. К.: НАУ. 2012. – С. 29 – 36

17. Мамон Е.В. Деякі аспекти захисту авторських прав у мережі Інтернет – [Електронний документ] – Режим доступу: http://www.yurlex.com.ua

18. Кінзерявий, О.М. Стеганографічні методи приховування даних у векторні зображення, стійкі до активних атак на основі афінних перетворень: дис. канд. техн.. наук. Спеціальність 05.13.21 – Системи захисту інформації. Київ , 2015, 324 с.

19. V. Patidar, N. K. Pareek, G. Purohit, K. K. Sud, “A robust and secure chaotic standard map based pseudorandom permutationsubstitution scheme for image encryption,” Opt. Commun., vol. 284, pp. 4331–4339, 2011.

20. Gutierrez-Cardenas J.M. Steganography and data loss prevention: an overlooked risk? // International Journal of Security and Its Applications. 2017. V. 11. N 4. P. 71–84.

21. J. Fridrich, “Image encryption based on chaotic maps,” in Proc. ICSMC, Orlando, FL, USA, USA, 1997, pp. 1105-1110.

22. Masilamani, “An efficient visually meaningful image encryption using Arnold transform,” in Proc. TechSym, Kharagpur, India, 2016, pp. 266-271.

23. l-Qahtani A, Tabakh A, Gutub A. 2009 Triple-A : secure RGB image steganography based onrandomization. In 7th ACS/IEEE Int. Conf.on Computer Systems and Applications (AICCSA-2009),Rabat, Morocco, pp.

24. Кузнецов О.О. Стеганография : навч. посібник / О.О. Кузнецов, С.П. Євсеєв, О.Г. Король. –Х. : Вид. ХНЕУ, 2011. –232 с

25. Prema C, Manimegalai D. 2014 Adaptive color image steganography using intra color pixel value differencing.Aust.J.BasicAppl.Sci.8

26. Gutub A. A. 2010 Pixel indicator technique for RGB image steganography. J. Emerg.Technol.WebIntell.(JETWI)

27. Parvez MT, Gutub AA. 2011 Vibrant color imagesteganography using channel differences and secretdata distribution.KuwaitJ.Sci.Eng

28. Gutierrez-Cardenas J.M. Steganography and data loss prevention: an overlooked risk? // International Journal of Security and Its Applications. 2017. V. 11. N 4. P. 71–84.

29. Nagaraj V, Vijayalakshmi V, Zayaraz G. 2013 Colourimage steganography based on pixel valuemodification method using modulus function. In2013Int.Conf.onElectronicEngineeringandComputerScienc

30. Miri A., Faez K. An image steganography method based on integer wavelet transform. Multimedia Tools and Applications. 2018. Vol. 77 (11). P 13133–13144.

31. Порівняння стійкості стеганографічних методів до різних типів спотворення програмними засобами | Системи обробки інформації. Наукові видання. URL: https://journal-hnups.com.ua/index.php/soi/article/view/185

32. Кущ С. М. Виявлення прихованих повідомлень як складова комплексних систем захисту інформації

33. Вступ в C#. programm. top: веб-сайт. URL: https://programm.top/uk/c-sharp/tutorial/introduction/ (дата звернення: 24.10.2022).

34. What is Windows Forms – Windows Forms .NET. Microsoft Learn: Build skills URL: https://learn.microsoft.com/en-us/dotnet/desktop/winforms

35. Visual Studio: IDE and Code Editor for Software Developers and Teams. Visual Studio. URL: https://visualstudio.microsoft.com/en/

36. NET Framework Microsoft. веб-сайт. URL: https://support.microsoft.com/microsoft-net-framework-9d23f658-3b97-68ab-d013-aa3c3e7495e0

37. Інформатика в прикладах – Основні компоненти програми для ОС з графічним інтерфейсом. Інформатика в прикладах – Головна. URL: http://nikolay.in.ua/distantsijne-navchannya/8-klas/840-osnovni-komponenti-programi-dlya-os-z-grafichnim-

38. Wikiwand – Графічний інтерфейс користувача. Wikiwand. URL: https://www.wikiwand.com/uk/Графічний_інтерфейс

39. Графічний інтерфейс. Кафедра математичної фізики | Новини. URL: http://www.matfiz.univ.kiev.ua/userfiles/files/Pres20_cm.pdf

40. В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт, ВНТУ, 2021

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції "Управління
інформаційною
безпекою" кафедри МБІС
д.т.н., професор

Юрій ЯРЕМЧУК

20 вересня 2023р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:
«Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення»

08-72.МКР.018.00.098.ТЗ

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент

Карпінець В.В.

Вінниця – 2023 р.

1. Найменування та область застосування

Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення, а також розроблення алгоритму відповідно до теми магістерської кваліфікаційної роботи.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №247 від 18. вересня 2023 р.

3. Мета та призначення розробки

Мета роботи у приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення, який підвищить рівень якості зображення після вбудовування ЦВЗ для захисту авторського права.

4. Джерела розробки

4.1. Конахович Г. Ф., Пузиренко А. Ю. Комп'ютерна стеганографія теорія і практика – Київ: МК-Прес. -2006.

4.2. Різуненко А. О. Теорія та практика цифрової обробки зображень – Полтава: РВВ ПУСКУ. – 2009.

4.3. Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп’ютерної стеганографії: Навчальни. посібник для студентів і аспірантів. – Вінниця: ВДТУ, 2003.

4.4. Po-Yueh Chen and Hung-Ju Lin “A DWT Based Approach for Image Steganography”, International Journal of Applied Science and Engineering, 2006. 4, 3: 275-290.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.2

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми	20.09.2023	25.09.2023
2	Аналіз предметної області обраної теми	26.09.2023	30.10.2023
3	Апробація отриманих результатів	31.10.2023	02.10.2023
4	Розробка алгоритму роботи	03.10.2023	17.10.2023
5	Написання магістерської роботи на основі розробленої теми	18.10.2023	10.11.2023
6	Розробка економічної частини	11.11.2023	23.11.2023
7	Передзахист магістерської кваліфікаційної роботи	24.11.2023	25.11.2023
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2023	30.11.2023
9	Захист магістерської кваліфікаційної роботи	15.12.2023	15.12.2023

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відзив керівника роботи;

– відзив опонента

Технічне завдання до виконання прийняв _____ Чечелюк О.В.

Додаток Б. Лістинг програми

```
using System;
using System.Drawing;
using System.Text;

class ImageProcessing
{
   // Green color range in HSV
   private static readonly int GreenLowerHue = 30;
   private static readonly int GreenLowerSaturation = 40;
   private static readonly int GreenLowerValue = 40;

   private static readonly int GreenUpperHue = 90;
   private static readonly int GreenUpperSaturation = 255;
   private static readonly int GreenUpperValue = 255;

   static void Main()
   {
      // Load the image
      Bitmap originalImage = new Bitmap("path/to/your/image.jpg");

      // Convert the image to HSV
      Bitmap hsvImage = ConvertToHSV(originalImage);

      // Define ranges for different objects
      var objects = new[]
      {
         new { Name = "Рослини", HueMin = 30, HueMax = 90, SaturationMin = 40,
               SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
         new { Name = "Небо", HueMin = 210, HueMax = 260, SaturationMin = 40,
               SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
         new { Name = "Пісок на пляжі", HueMin = 20, HueMax = 40, SaturationMin =
               40, SaturationMax = 255, ValueMin = 40, ValueMax = 255 },
         new { Name = "Сніг", HueMin = 0, HueMax = 30, SaturationMin = 0,
               SaturationMax = 40, ValueMin = 200, ValueMax = 255 },
         new { Name = "Гірські вершини", HueMin = 20, HueMax = 40, SaturationMin =
               40, SaturationMax = 255, ValueMin = 40, ValueMax = 255 }
      };

      // Detect objects and get coordinates
      foreach (var obj in objects)
      {
         List<Tuple<int, int>> objectCoordinates =
               DetectObjectCoordinates(hsvImage, obj);
         // Output the result for each object
         Console.WriteLine($"Detected {obj.Name} at coordinates:");
         foreach (var coordinate in objectCoordinates)
         {
            Console.WriteLine($"X: {coordinate.Item1}, Y: {coordinate.Item2}");
         }
         Console.WriteLine();
```

```
    }
}

 static Bitmap ConvertToHSV(Bitmap originalImage)
 {
    int width = originalImage.Width;
    int height = originalImage.Height;

    // Create a new bitmap for the HSV image
    Bitmap hsvImage = new Bitmap(width, height);

    for (int i = 0; i < width; i++)
    {
       for (int j = 0; j < height; j++)
       {
          // Get the color of the pixel in the original image
          Color originalColor = originalImage.GetPixel(i, j);

          // Convert the color to HSV
          Color hsvColor = RGBtoHSV(originalColor);

          // Set the corresponding pixel in the new HSV image
          hsvImage.SetPixel(i, j, hsvColor);
       }
    }

    return hsvImage;
 }

 static Bitmap ConvertToRGB(Bitmap hsvImage)
 {
    int width = hsvImage.Width;
    int height = hsvImage.Height;

    // Create a new bitmap for the RGB image
    Bitmap rgbImage = new Bitmap(width, height);

    for (int i = 0; i < width; i++)
    {
       for (int j = 0; j < height; j++)
       {
          // Get the color of the pixel in the HSV image
          Color hsvColor = hsvImage.GetPixel(i, j);

          // Convert the color to RGB
          Color rgbColor = HSVtoRGB(hsvColor);

          // Set the corresponding pixel in the new RGB image
          rgbImage.SetPixel(i, j, rgbColor);
       }
    }
```

```
    return rgbImage;
}

static List<Tuple<int, int>> DetectObjectCoordinates(Bitmap hsvImage, object obj)
{
    List<Tuple<int, int>> coordinates = new List<Tuple<int, int>>();

    // Iterate through each pixel in the image
    for (int x = 0; x < hsvImage.Width; x++)
    {
        for (int y = 0; y < hsvImage.Height; y++)
        {
            // Get the HSV values of the current pixel
            Color hsvColor = hsvImage.GetPixel(x, y);
            int pixelHue = (int)(hsvColor.GetHue() * 2);
            int pixelSaturation = (int)(hsvColor.GetSaturation() * 255);
            int pixelValue = (int)(hsvColor.GetBrightness() * 255);

            // Check if the pixel falls within the specified ranges for the object
            if (IsPixelInObjectRange(pixelHue, pixelSaturation, pixelValue, obj))
            {
                coordinates.Add(new Tuple<int, int>(x, y));
            }
        }
    }

    return coordinates;
}

static bool IsPixelInObjectRange(int hue, int saturation, int value, object obj)
{
    return hue >= obj.HueMin && hue <= obj.HueMax &&
           saturation >= obj.SaturationMin && saturation <= obj.SaturationMax &&
           value >= obj.ValueMin && value <= obj.ValueMax;
}
static void HaarWaveletTransform2D(Bitmap image)
{
    int width = image.Width;
    int height = image.Height;

    // Transform each row
    for (int i = 0; i < height; i++)
    {
        HaarWaveletTransform(image, i, 0, width, 1);
    }

    // Transform each column
    for (int j = 0; j < width; j++)
    {
        HaarWaveletTransform(image, 0, j, height, width);
    }
}
```

```
        static void HaarWaveletTransform(Bitmap image, int startRow, int startCol, int length,
int stride)
        {
            for (int step = length / 2; step >= 1; step /= 2)
            {
                for (int i = 0; i < step; i++)
                {
                    int x = startRow + i * 2;
                    for (int j = startCol; j < startCol + stride; j++)
                    {
                        double sum = image.GetPixel(x, j).GetBrightness() + image.GetPixel(x + 1,
j).GetBrightness();
                        double diff = image.GetPixel(x, j).GetBrightness() - image.GetPixel(x + 1,
j).GetBrightness();

                        Color sumColor = ColorFromBrightness(sum);
                        Color diffColor = ColorFromBrightness(diff);

                        image.SetPixel(x / 2, j, sumColor);
                        image.SetPixel(x / 2 + step, j, diffColor);
                    }
                }
            }
        }

        static void InverseHaarWaveletTransform2D(Bitmap image)
        {
            int width = image.Width;
            int height = image.Height;

            // Inverse transform each column
            for (int j = 0; j < width; j++)
            {
                InverseHaarWaveletTransform(image, 0, j, height, width);
            }

            // Inverse transform each row
            for (int i = 0; i < height; i++)
            {
                InverseHaarWaveletTransform(image, i, 0, width, 1);
            }
        }

        static void InverseHaarWaveletTransform(Bitmap image, int startRow, int startCol, int
length, int stride)
        {
            for (int step = 1; step <= length / 2; step *= 2)
            {
                for (int i = 0; i < step; i++)
                {
                    int x = startRow + i * 2;
```

```
            for (int j = startCol; j < startCol + stride; j++)
            {
                Color sumColor = image.GetPixel(x / 2, j);
                Color diffColor = image.GetPixel(x / 2 + step, j);

                double sum = sumColor.GetBrightness();
                double diff = diffColor.GetBrightness();

                double avg = (sum + diff) / 2;
                double delta = (sum - diff) / 2;

                image.SetPixel(x, j, ColorFromBrightness(avg + delta));
                image.SetPixel(x + 1, j, ColorFromBrightness(avg - delta));
            }
        }
    }
}

static int AnalyzeGreenPixels(Bitmap greenChannel)
{
    int width = greenChannel.Width;
    int height = greenChannel.Height;

    // Area of green pixels in the reconstructed green channel
    int greenArea = 0;

    for (int i = 0; i < width; i++)
    {
        for (int j = 0; j < height; j++)
        {
            // Get the color of the pixel in the reconstructed green channel
            Color pixelColor = greenChannel.GetPixel(i, j);

            // Check if the pixel is green
            if (pixelColor.G > 0) // Assuming non-zero value indicates green
            {
                greenArea++;
            }
        }
    }

    return greenArea;
}

static bool IsGreenPixel(Color hsvColor)
{
    int hue = hsvColor.R;
    int saturation = hsvColor.G;
    int value = hsvColor.B;

    return (hue >= GreenLowerHue && hue <= GreenUpperHue &&
```

```
            saturation >= GreenLowerSaturation && saturation <= GreenUpperSaturation
&&
            value >= GreenLowerValue && value <= GreenUpperValue);
        }

        static Color RGBtoHSV(Color rgbColor)
        {
            float r = rgbColor.R / 255.0f;
            float g = rgbColor.G / 255.0f;
            float b = rgbColor.B / 255.0f;

            float max = Math.Max(r, Math.Max(g, b));
            float min = Math.Min(r, Math.Min(g, b));

            float h, s, v;

            // Hue calculation
            if (max == min)
            {
                h = 0; // undefined
            }
            else if (max == r)
            {
                h = (60 * (g - b) / (max - min) + 360) % 360;
            }
            else if (max == g)
            {
                h = (60 * (b - r) / (max - min) + 120);
            }
            else // max == b
            {
                h = (60 * (r - g) / (max - min) + 240);
            }

            // Saturation calculation
            if (max == 0)
            {
                s = 0;
            }
            else
            {
                s = (max - min) / max;
            }

            // Value calculation
            v = max;

            // Convert hue to 0-255 range
            h = (h / 360) * 255;

            return Color.FromArgb((int)h, (int)(s * 255), (int)(v * 255));
        }
```

```
static Color ColorFromBrightness(double brightness)
{
   int value = (int)(brightness * 255);
   return Color.FromArgb(value, value, value);
}

static Color HSVtoRGB(Color hsvColor)
{
   float h = hsvColor.R * 360.0f / 255.0f;
   float s = hsvColor.G / 255.0f;
   float v = hsvColor.B / 255.0f;

   int hi = Convert.ToInt32(Math.Floor(h / 60)) % 6;
   float f = (h / 60) - Math.Floor(h / 60);

   float p = v * (1 - s);
   float q = v * (1 - f * s);
   float t = v * (1 - (1 - f) * s);

   switch (hi)
   {
      case 0:
         return Color.FromArgb((int)(v * 255), (int)(t * 255), (int)(p * 255));
      case 1:
         return Color.FromArgb((int)(q * 255), (int)(v * 255), (int)(p * 255));
      case 2:
         return Color.FromArgb((int)(p * 255), (int)(v * 255), (int)(t * 255));
      case 3:
         return Color.FromArgb((int)(p * 255), (int)(q * 255), (int)(v * 255));
      case 4:
         return Color.FromArgb((int)(t * 255), (int)(p * 255), (int)(v * 255));
      default:
         return Color.FromArgb((int)(v * 255), (int)(p * 255), (int)(q * 255));
   }
}

static void EncodeSecretMessage(Bitmap greenChannel, string message)
{
   byte[] messageBytes = Encoding.UTF8.GetBytes(message);

   int width = greenChannel.Width;
   int height = greenChannel.Height;

   int messageIndex = 0;

   for (int i = 0; i < width; i++)
   {
      for (int j = 0; j < height; j++)
      {
         // Get the color of the pixel in the DWT-transformed green channel
         Color pixelColor = greenChannel.GetPixel(i, j);
```

```
                    // Encode the message in the green channel value
                    int greenValue = pixelColor.G;
                    if (messageIndex < messageBytes.Length)
                    {
                        greenValue = (greenValue & 0xFE) | ((messageBytes[messageIndex] >> 7)
& 0x01);
                        messageIndex++;
                    }

                    greenChannel.SetPixel(i, j, Color.FromArgb(greenValue, greenValue,
greenValue));
                }
            }
        }
    }


    namespace HSVandDWT
    {
        partial class Form1
        {
            /// <summary>
            /// Required designer variable.
            /// </summary>
            private System.ComponentModel.IContainer components = null;

            /// <summary>
            /// Clean up any resources being used.
            /// </summary>
            /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
            protected override void Dispose(bool disposing)
            {
                if (disposing && (components != null))
                {
                    components.Dispose();
                }
                base.Dispose(disposing);
            }

            #region Windows Form Designer generated code

            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
                this.tabPanel = new System.Windows.Forms.TabControl();
                this.encodePage = new System.Windows.Forms.TabPage();
                this.choseImageForEncode = new System.Windows.Forms.Label();
```

```
            this.encodePictureBox = new System.Windows.Forms.PictureBox();
            this.encodeLabel = new System.Windows.Forms.Label();
            this.encodeTextBox1 = new System.Windows.Forms.TextBox();
            this.encodeButton = new System.Windows.Forms.Button();
            this.decodePage = new System.Windows.Forms.TabPage();
            this.decodePictureBox = new System.Windows.Forms.PictureBox();
            this.choseImageForDecode = new System.Windows.Forms.Label();
            this.decodeButton = new System.Windows.Forms.Button();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.xLabel = new System.Windows.Forms.Label();
            this.yLabel = new System.Windows.Forms.Label();
            this.xTextBox = new System.Windows.Forms.TextBox();
            this.yTextBox = new System.Windows.Forms.TextBox();
            this.heightLabel = new System.Windows.Forms.Label();
            this.widthLabel = new System.Windows.Forms.Label();
            this.heightTextBox = new System.Windows.Forms.TextBox();
            this.widthTextBox = new System.Windows.Forms.TextBox();
            this.tabPanel.SuspendLayout();
            this.encodePage.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.encodePictureBox)).BeginInit();
            this.decodePage.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.decodePictureBox)).BeginInit();
            this.groupBox1.SuspendLayout();
            this.SuspendLayout();
            //
            // tabPanel
            //
            this.tabPanel.Anchor = System.Windows.Forms.AnchorStyles.None;
            this.tabPanel.Controls.Add(this.encodePage);
            this.tabPanel.Controls.Add(this.decodePage);
            this.tabPanel.Location = new System.Drawing.Point(1, 0);
            this.tabPanel.Name = "tabPanel";
            this.tabPanel.SelectedIndex = 0;
            this.tabPanel.Size = new System.Drawing.Size(480, 450);
            this.tabPanel.TabIndex = 0;
            //
            // encodePage
            //
            this.encodePage.AutoScroll = true;
            this.encodePage.Controls.Add(this.choseImageForEncode);
            this.encodePage.Controls.Add(this.encodePictureBox);
            this.encodePage.Controls.Add(this.encodeLabel);
            this.encodePage.Controls.Add(this.encodeTextBox1);
            this.encodePage.Controls.Add(this.encodeButton);
            this.encodePage.Location = new System.Drawing.Point(4, 22);
            this.encodePage.Name = "encodePage";
            this.encodePage.Padding = new System.Windows.Forms.Padding(3);
            this.encodePage.Size = new System.Drawing.Size(472, 424);
            this.encodePage.TabIndex = 0;
            this.encodePage.Text = "Encode";
```

```
            this.encodePage.UseVisualStyleBackColor = true;
            //
            // choseImageForEncode
            //
            this.choseImageForEncode.AutoSize = true;
            this.choseImageForEncode.Location = new System.Drawing.Point(10, 7);
            this.choseImageForEncode.Name = "choseImageForEncode";
            this.choseImageForEncode.Size = new System.Drawing.Size(126, 13);
            this.choseImageForEncode.TabIndex = 4;
            this.choseImageForEncode.Text = "Chose image for Encode:";
            //
            // encodePictureBox
            //
            this.encodePictureBox.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
            | System.Windows.Forms.AnchorStyles.Left)
            | System.Windows.Forms.AnchorStyles.Right)));
            this.encodePictureBox.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
            this.encodePictureBox.Location = new System.Drawing.Point(140, 6);
            this.encodePictureBox.Name = "encodePictureBox";
            this.encodePictureBox.Size = new System.Drawing.Size(200, 200);
            this.encodePictureBox.TabIndex = 3;
            this.encodePictureBox.TabStop = false;
            this.encodePictureBox.Click += new
System.EventHandler(this.encodePictureBox_Click);
            //
            // encodeLabel
            //
            this.encodeLabel.AutoSize = true;
            this.encodeLabel.Location = new System.Drawing.Point(7, 215);
            this.encodeLabel.Name = "encodeLabel";
            this.encodeLabel.Size = new System.Drawing.Size(144, 13);
            this.encodeLabel.TabIndex = 2;
            this.encodeLabel.Text = "Type a secret message here:";
            //
            // encodeTextBox1
            //
            this.encodeTextBox1.Location = new System.Drawing.Point(7, 231);
            this.encodeTextBox1.Multiline = true;
            this.encodeTextBox1.Name = "encodeTextBox1";
            this.encodeTextBox1.Size = new System.Drawing.Size(456, 107);
            this.encodeTextBox1.TabIndex = 1;
            //
            // encodeButton
            //
            this.encodeButton.Location = new System.Drawing.Point(7, 344);
            this.encodeButton.Name = "encodeButton";
            this.encodeButton.Size = new System.Drawing.Size(456, 72);
            this.encodeButton.TabIndex = 0;
            this.encodeButton.Text = "Encode";
```

```
            this.encodeButton.UseVisualStyleBackColor = true;
            this.encodeButton.Click += new System.EventHandler(this.encodeButton_Click);
            //
            // decodePage
            //
            this.decodePage.Controls.Add(this.groupBox1);
            this.decodePage.Controls.Add(this.decodePictureBox);
            this.decodePage.Controls.Add(this.choseImageForDecode);
            this.decodePage.Controls.Add(this.decodeButton);
            this.decodePage.Location = new System.Drawing.Point(4, 22);
            this.decodePage.Name = "decodePage";
            this.decodePage.Padding = new System.Windows.Forms.Padding(3);
            this.decodePage.Size = new System.Drawing.Size(472, 424);
            this.decodePage.TabIndex = 1;
            this.decodePage.Text = "Decode";
            this.decodePage.UseVisualStyleBackColor = true;
            //
            // decodePictureBox
            //
            this.decodePictureBox.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
            this.decodePictureBox.Location = new System.Drawing.Point(140, 6);
            this.decodePictureBox.Name = "decodePictureBox";
            this.decodePictureBox.Size = new System.Drawing.Size(200, 200);
            this.decodePictureBox.TabIndex = 2;
            this.decodePictureBox.TabStop = false;
            this.decodePictureBox.Click += new
System.EventHandler(this.decodePictureBox_Click);
            //
            // choseImageForDecode
            //
            this.choseImageForDecode.AutoSize = true;
            this.choseImageForDecode.Location = new System.Drawing.Point(10, 7);
            this.choseImageForDecode.Name = "choseImageForDecode";
            this.choseImageForDecode.Size = new System.Drawing.Size(127, 13);
            this.choseImageForDecode.TabIndex = 1;
            this.choseImageForDecode.Text = "Chose image for Decode:";
            //
            // decodeButton
            //
            this.decodeButton.Location = new System.Drawing.Point(7, 344);
            this.decodeButton.Name = "decodeButton";
            this.decodeButton.Size = new System.Drawing.Size(456, 72);
            this.decodeButton.TabIndex = 0;
            this.decodeButton.Text = "Decode";
            this.decodeButton.UseVisualStyleBackColor = true;
            this.decodeButton.Click += new System.EventHandler(this.decodeButton_Click);
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.widthTextBox);
            this.groupBox1.Controls.Add(this.heightTextBox);
```

```
this.groupBox1.Controls.Add(this.widthLabel);
this.groupBox1.Controls.Add(this.heightLabel);
this.groupBox1.Controls.Add(this.yTextBox);
this.groupBox1.Controls.Add(this.xTextBox);
this.groupBox1.Controls.Add(this.yLabel);
this.groupBox1.Controls.Add(this.xLabel);
this.groupBox1.Location = new System.Drawing.Point(7, 212);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(456, 126);
this.groupBox1.TabIndex = 3;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Chose position";
//
// xLabel
//
this.xLabel.AutoSize = true;
this.xLabel.Location = new System.Drawing.Point(7, 31);
this.xLabel.Name = "xLabel";
this.xLabel.Size = new System.Drawing.Size(17, 13);
this.xLabel.TabIndex = 0;
this.xLabel.Text = "X:";
//
// yLabel
//
this.yLabel.AutoSize = true;
this.yLabel.Location = new System.Drawing.Point(8, 78);
this.yLabel.Name = "yLabel";
this.yLabel.Size = new System.Drawing.Size(17, 13);
this.yLabel.TabIndex = 1;
this.yLabel.Text = "Y:";
//
// xTextBox
//
this.xTextBox.Location = new System.Drawing.Point(31, 28);
this.xTextBox.Name = "xTextBox";
this.xTextBox.Size = new System.Drawing.Size(100, 20);
this.xTextBox.TabIndex = 2;
//
// yTextBox
//
this.yTextBox.Location = new System.Drawing.Point(31, 75);
this.yTextBox.Name = "yTextBox";
this.yTextBox.Size = new System.Drawing.Size(100, 20);
this.yTextBox.TabIndex = 3;
//
// heightLabel
//
this.heightLabel.AutoSize = true;
this.heightLabel.Location = new System.Drawing.Point(269, 31);
this.heightLabel.Name = "heightLabel";
this.heightLabel.Size = new System.Drawing.Size(41, 13);
this.heightLabel.TabIndex = 4;
```

```
    this.heightLabel.Text = "Height:";
    //
    // widthLabel
    //
    this.widthLabel.AutoSize = true;
    this.widthLabel.Location = new System.Drawing.Point(272, 77);
    this.widthLabel.Name = "widthLabel";
    this.widthLabel.Size = new System.Drawing.Size(38, 13);
    this.widthLabel.TabIndex = 5;
    this.widthLabel.Text = "Width:";
    //
    // heightTextBox
    //
    this.heightTextBox.Location = new System.Drawing.Point(316, 28);
    this.heightTextBox.Name = "heightTextBox";
    this.heightTextBox.Size = new System.Drawing.Size(100, 20);
    this.heightTextBox.TabIndex = 6;
    //
    // widthTextBox
    //
    this.widthTextBox.Location = new System.Drawing.Point(316, 74);
    this.widthTextBox.Name = "widthTextBox";
    this.widthTextBox.Size = new System.Drawing.Size(100, 20);
    this.widthTextBox.TabIndex = 7;
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.BackColor = System.Drawing.SystemColors.Control;
    this.ClientSize = new System.Drawing.Size(480, 450);
    this.Controls.Add(this.tabPanel);
    this.Name = "Form1";
    this.Text = "Steganography: HSV and DWT";
    this.Load += new System.EventHandler(this.Form1_Load);
    this.tabPanel.ResumeLayout(false);
    this.encodePage.ResumeLayout(false);
    this.encodePage.PerformLayout();
    ((System.ComponentModel.ISupportInitialize)(this.encodePictureBox)).EndInit();
    this.decodePage.ResumeLayout(false);
    this.decodePage.PerformLayout();
    ((System.ComponentModel.ISupportInitialize)(this.decodePictureBox)).EndInit();
    this.groupBox1.ResumeLayout(false);
    this.groupBox1.PerformLayout();
    this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.TabControl tabPanel;
private System.Windows.Forms.TabPage encodePage;
```

```
        private System.Windows.Forms.TabPage decodePage;
        private System.Windows.Forms.Button encodeButton;
        private System.Windows.Forms.Button decodeButton;
        private System.Windows.Forms.TextBox encodeTextBox1;
        private System.Windows.Forms.Label encodeLabel;
        private System.Windows.Forms.Label choseImageForEncode;
        private System.Windows.Forms.PictureBox encodePictureBox;
        private System.Windows.Forms.Label choseImageForDecode;
        private System.Windows.Forms.PictureBox decodePictureBox;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.TextBox yTextBox;
        private System.Windows.Forms.TextBox xTextBox;
        private System.Windows.Forms.Label yLabel;
        private System.Windows.Forms.Label xLabel;
        private System.Windows.Forms.TextBox widthTextBox;
        private System.Windows.Forms.TextBox heightTextBox;
        private System.Windows.Forms.Label widthLabel;
        private System.Windows.Forms.Label heightLabel;
    }
}
```

Додаток Г. Ілюстративний матеріал

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення

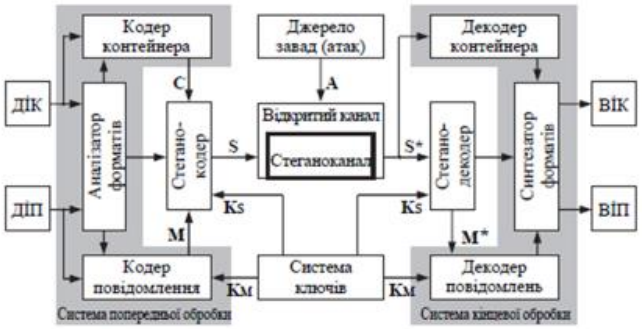
ВИКОНАВ: СТ. 2-ГО КУРСУ ГРУПИ 2КІТС-22М
ЧЕЧЕЛЮК ОЛЕКСАНДР ВАСИЛЬОВИЧ
КЕРІВНИК: К.Т.Н., ДОЦ., ДОЦЕНТ КАФ. МБІС
КАРПІНЕЦЬ ВАСИЛЬ ВАСИЛЬОВИЧ

Актуальність та новизна роботи

Метою роботи є підвищення стійкості методу дискретного вейвлет-перетворення шляхом розширення областей вбудовування

Наукова новизна: вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення шляхом розширення областей вбудовування

Компоненти типової стеганосистеми



Формула дискретного вейвлет-перетворення

$$W(a,b) = \sum_n x[n] \cdot \psi^* \left(\frac{n-a}{a}\right)$$

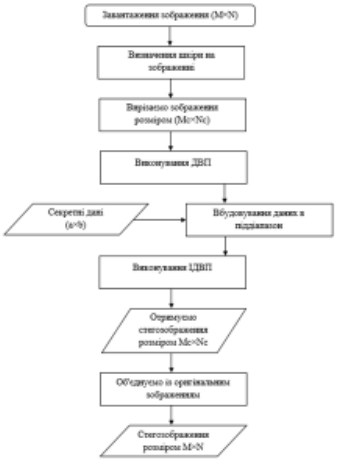
$x[n]$ - вхідний сигнал;

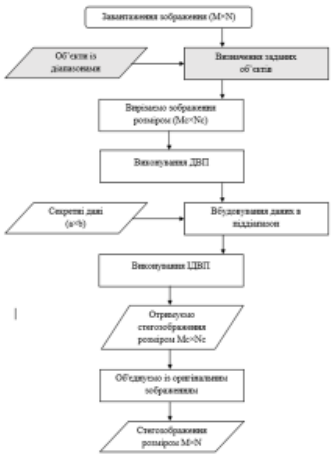
$\psi[n]$ - вейвлет-функція;

a - масштабний коефіцієнт;

b – зсув;

Розробка вдосконаленого методу для приховування інформації





Діапазон кольорів для об’єктів

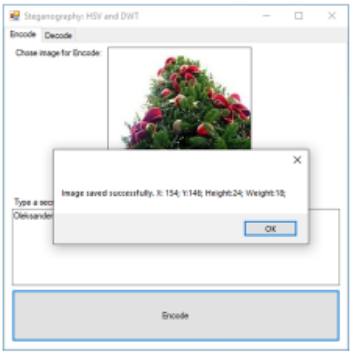
Об'єкт	Нижній межа	Верхня межа
Рослини	30, 40, 40	90, 255, 255
Небо	210, 40, 40	260, 255, 255
Пісок на пляжі	20, 40, 40	40, 255, 255
Сніг	0, 0, 200	30, 40, 255
Гірські вершини	20, 40, 40	40, 255, 255

Вибір засобів програмування

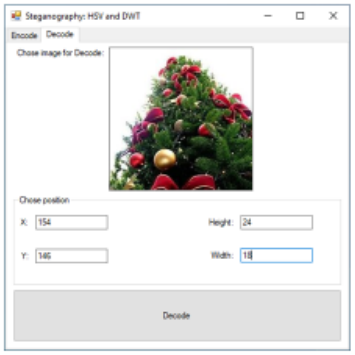
Виходячи із поставлених завдань роботи, для програмного додатку обрано наступні **програмні засоби**:

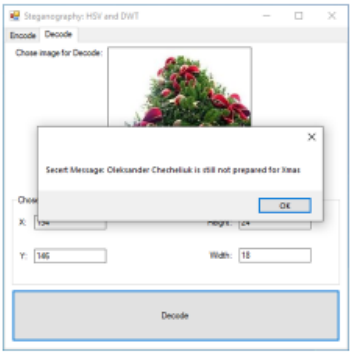
- мова об'єктно-орієнтовного програмування **C#;**
- середовище програмування **Visual Studio 2022;**
- інтерфейс програмування додатків **Windows Forms;**
- програмна технологія для створення додатків **.Net Framework.**

Інтерфейс програмного додатку



Інтерфейс програмного додатку





Аналіз стійкості вдосконаленого методу

Назва методу	Зображення	PSNR (дБ)	MSE	Ємність (біт)
Базовий	Дитина	69.5104	0.00727844	1248890
Вдосконалений	Рослина	70.3077	0.00605776	1225240
Вдосконалений	Небо	68.3502	0.00734539	1238770
Вдосконалений	Пісок на пляжі	70.1243	0.00648375	1215370
Вдосконалений	Сніг	69.4212	0.00675345	1256870
Вдосконалений	Гірські вершини	69.8043	0.00743641	1268763

Економічна доцільність розробки

Проведено дослідження економічного потенціалу розробки, зокрема, оцінювання комерційного потенціалу; прогнозування витрат на виконання наукової роботи та впровадження її результатів; прогнозування комерційних ефектів від реалізації результатів розробки та розрахунок ефективності вкладених інвестицій та період їх окупності.

В результаті аналізу отриманих економічних показників, можна вважати, що запропонована розробка програмного засобу з використанням алгоритму для вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

Висновки

Враховуючи актуальність обраної теми, в роботі було **здійснено підвищення стійкості методу дискретного вейвлет-перетворення шляхом розширення областей вбудовування.**

Розроблено програмний додаток на основі вдосконаленого методі. Отриманий результат свідчить про успішність вдосконаленого методу та доцільність його застосування на практиці.

Дякую за увагу!

Додаток Д . Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ
РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ
ЗАПОЗИЧЕНЬ

Назва роботи: Вдосконалення методу приховування інформації на основі колірної моделі HSV та дискретного вейвлет-перетворення

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки

(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 84 % Схожість 16 %

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Чечелюк О.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Карпінець В.В.
(підпис) (прізвище, ініціали)