Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет менеджменту та інформаційної безпеки
(повне найменування інституту, назва факультету (відділення))

Кафедра менеджменту та безпеки інформаційних систем
(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота на тему:

«ПІДВИЩЕННЯ СТІЙКОСТІ МЕТОДУ ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯХ ДО ПАСИВНИХ АТАК НА ОСНОВІ ВИСОКОЧАСТОТНИХ КОЕФІЦІЄНТІВ DWT ТА КОДІВ КОРЕКЦІЇ ПОМИЛОК»

Виконав: студент 2 курсу, групи
2КІТС-24м спеціальності 125 –
Кібербезпека та захист інформації
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)
Костюк І.А.
(прізвище та ініціали)

Керівник: к.т.н., доц., зав. каф. МБІС
Карпінець В. В.
(прізвище та ініціали)
«11» грудня 2025 р.

Опонент: к.т.н., доц., доц. каф. ОТ
Тарновський М. Г.
(прізвище та ініціали)
«11» грудня 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
Юрій ЯРЕМЧУК
“11” грудня 2025 р.

Вінниця ВНТУ - 2025 рік

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

Рівень вищої освіти ІІ-ий (магістерський)
Спеціальність 125 – Кібербезпека та захист інформації
Освітньо-професійна програма – Кібербезпека інформаційних технологій та систем.

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____**Юрій ЯРЕМЧУК**
"24" вересня 2025 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Костюку Іллі Андрійовичу

1. Тема роботи: «Підвищення стійкості методу приховування інформації у зображеннях до пасивних атак на основі високочастотних коефіцієнтів dwt та кодів корекції помилок»
2. Керівник роботи: к.т.н., доц., зав. каф. МБІС Карпінець В.В.
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "24" вересня 2025 року № 313
3. Строк подання студентом роботи: 01.12.2025 р.
4. Вихідні дані до роботи: Методичні вказівки до виконання магістерської кваліфікаційної роботи; наукові публікації та монографії з питань цифрової стеганографії та обробки цифрових зображень; математичний опис дискретного вейвлет-перетворення (DWT); теорія завадостійкого кодування та кодів корекції помилок (Ріда-Соломона); специфікації стандартів стиснення зображень (JPEG).
5. Зміст текстової частини:
1. Аналіз сучасних методів стеганографії та стегоаналізу цифрових зображень
2. Розроблення методу підвищення стійкості приховування інформації у зображеннях до пасивних атак на основі високочастотних коефіцієнтів dwt та кодів корекції помилок
3. Експериментальне дослідження та порівняльний аналіз розробленого методу
4. Економічна частина
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
6. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень): Класифікація методів стеганографії та стегоаналізу, Структурна схема розробленого методу DWT+ECC, Алгоритми вбудовування та витягування інформації, Програмна реалізація та інтерфейс системи,

Результати експериментального дослідження стійкості та непомітності, економічн показники ефективності розробки

7. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	к.т.н., доц., зав. каф. МБІС Карпінець В.В.		
Економічна частина	к.т.н., доц. каф. ЕПВМ Ратушняк О.Г.		

8. Дата видачі завдання: 24 вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Отримання завдання, підбір та аналіз літературних джерел з питань стеганографії цифрових зображень	24.09.2025	05.10.2025	
2	Аналіз існуючих рішень та обґрунтування вибору DWT-перетворення і кодів корекції помилок	06.10.2025	15.10.2025	
3	Розробка математичної моделі та алгоритму вбудовування даних у високочастотні коефіцієнти DWT	16.10.2025	28.10.2025	
4	Програмна реалізація методу та проведення експериментальних досліджень стійкості до атак	29.10.2025	14.11.2025	
5	Підготовка економічної частини та розрахунок ефективності розробки	15.11.2025	19.11.2025	
6	Оформлення пояснювальної записки, підготовка графічного матеріалу та презентаці	20.11.2025	26.11.2025	
7	Переддипломний захист	27.11.2025	28.11.2025	
8	Захист магістерської кваліфікаційної роботи	08.12.2025	11.12.2025	

Студент _____ **Костюк І. А.**
(підпис) (прізвище та ініціали)

Керівник роботи _____ **Карпінець В. В.**
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.492.3

Костюк І. А. Підвищення стійкості методу приховування інформації у зображеннях до пасивних атак на основі високочастотних коефіцієнтів DWT та кодів корекції помилок. Магістерська кваліфікаційна робота зі спеціальності 125 – кібербезпека, освітня програма – кібербезпека та захист інформаційних технологій і систем. Вінниця: ВНТУ, 2025. n с.

На укр. мові. Бібліогр.: 28 назв; рис.: 40; табл.: 21.

У магістерській кваліфікаційній роботі розроблено метод приховування інформації у цифрових зображеннях на основі високочастотних коефіцієнтів дискретного вейвлет-перетворення з використанням кодів корекції помилок. Метод дозволяє підвищити стійкість прихованих даних до пасивних атак, зокрема JPEG-стиснення, накладання шуму та фільтрації. У першому розділі проведено аналіз сучасних методів стеганографії у просторовій та частотній областях, досліджено підходи до підвищення стійкості до атак. У другому розділі обґрунтовано вибір високочастотних коефіцієнтів DWT для вбудовування даних, розроблено алгоритм приховування інформації з використанням кодів корекції помилок, запропоновано адаптивну стратегію вбудовування на основі текстурних характеристик зображення.

Графічна частина складається з n зображень із результатами моделювання та експериментального дослідження.

Ключові слова: стеганографія, дискретне вейвлет-перетворення, високочастотні коефіцієнти, коди корекції помилок, пасивні атаки, JPEG-стиснення, цифрові зображення.

ABSTRACT

UDC 004.492.3

Kostiuk I. A. Improving the robustness of information hiding method in images against passive attacks based on high-frequency DWT coefficients and error correction codes. Master's thesis in specialty 125 – cybersecurity, educational program – cybersecurity and protection of information technologies and systems. Vinnytsia: VNTU, 2025. – n p.

In Ukrainian language. Bibliographer: 28 titles; fig.: 40; tabl.: 21.

In the master's thesis, a method for hiding information in digital images based on high-frequency coefficients of discrete wavelet transform with the use of error correction codes has been developed. The method allows to increase the robustness of hidden data against passive attacks, particularly JPEG compression, noise addition and filtering. In the first chapter, an analysis of modern steganography methods in spatial and frequency domains has been conducted, approaches to improving robustness against attacks have been investigated. In the second chapter, the choice of high-frequency DWT coefficients for data embedding has been substantiated, an algorithm for information hiding using error correction codes has been developed, an adaptive embedding strategy based on texture characteristics of the image has been proposed.

The graphical part consists of n posters with simulation and experimental research results.

Keywords: steganography, discrete wavelet transform, high-frequency coefficients, error correction codes, passive attacks, JPEG compression, digital images.

ЗМІСТ

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій питання забезпечення конфіденційності, цілісності та прихованості інформації набуває особливої актуальності. Зростання обсягів мультимедійних даних, активне використання хмарних сервісів, соціальних мереж та глобальних систем комунікації створюють нові виклики у сфері інформаційної безпеки. Традиційні криптографічні методи, хоч і забезпечують високий рівень захисту, не маскують сам факт існування секретного повідомлення, що може привертати увагу зловмисників та підвищувати ризик атак. На цьому тлі стеганографія — наука про приховування факту передавання інформації у звичайних цифрових контейнерах — стає важливим інструментом сучасних систем кібербезпеки.

Цифрові зображення є одним із найбільш популярних та зручних контейнерів для приховування даних завдяки їх високій ємності, структурній складності та широкій поширеності у різноманітних комунікаційних середовищах. Водночас стеганографічні методи, що працюють безпосередньо у просторовій області, часто виявляються вразливими до пасивних атак, таких як JPEG-стиснення, фільтрація, додавання шуму або повторне кодування. Ці види обробки призводять до суттєвих спотворень у вбудованій інформації, що суттєво знижує практичну придатність класичних методів стеганографії.

Використання дискретного вейвлет-перетворення (DWT) відкриває можливості для побудови більш стійких методів приховування даних. DWT забезпечує багаторівневий аналіз структури зображення та розділяє його на компоненти різних частот, що дозволяє враховувати властивості людського зору та специфіку алгоритмів стискання. Особливий інтерес становлять високочастотні коефіцієнти, які хоча й є більш чутливими до шумів, проте можуть забезпечувати високу прихованість і контрольоване вбудовування, якщо доповнити їх застосуванням кодів корекції помилок. Поєднання цих

підходів створює основу для підвищення стійкості прихованої інформації до поширених пасивних атак.

Проблематика підвищення надійності стеганографічних методів є важливою для широкого спектра практичних застосувань: національної безпеки, захисту авторських прав, корпоративних комунікацій, медичної сфери, банківського сектору та інших галузей, де конфіденційність передавання інформації має принципове значення. Однією з ключових задач є досягнення оптимального балансу між непомітністю змін у контейнері, ємністю вбудованої інформації та стійкістю до впливу зовнішніх факторів. Саме тому дослідження методів вбудовування у частотну область, зокрема у високочастотні компоненти DWT з використанням кодів корекції помилок, є актуальним та науково значущим.

Метою магістерської роботи є розробка та дослідження методу приховування інформації у цифрових зображеннях на основі високочастотних коефіцієнтів дискретного вейвлет-перетворення із застосуванням кодів корекції помилок для забезпечення стійкості до пасивних атак. Об’єктом дослідження є процес приховування та вилучення конфіденційної інформації у цифрових зображеннях в умовах впливу пасивних атак.

Предметом дослідження є методи та алгоритми стеганографічного вбудовування даних у високочастотні коефіцієнти DWT із застосуванням адаптивних стратегій та кодів корекції помилок.

Об’єктом дослідження є процес приховування та відновлення конфіденційної інформації у цифрових зображеннях в умовах впливу пасивних атак, що виникають під час обробки, передачі та зберігання мультимедійних даних.

Наукова новизна одержаних результатів полягає у розробленні та теоретичному обґрунтуванні комбінованого методу приховування інформації у цифрових зображеннях, що поєднує використання високочастотних коефіцієнтів дискретного вейвлет-перетворення із

застосуванням кодів корекції помилок для забезпечення підвищеної стійкості до пасивних атак.

Для досягнення поставленої мети у роботі вирішуються такі завдання:

1) виконати аналіз сучасних методів стеганографії у просторовій та частотній областях;
2) дослідити підходи до підвищення стійкості стеганографічних систем до пасивних атак;
3) обґрунтувати вибір високочастотних коефіцієнтів DWT як середовища для вбудовування інформації;
4) розробити стеганографічний алгоритм із використанням кодів корекції помилок;
5) запропонувати адаптивну стратегію вбудовування, орієнтовану на текстурні характеристики зображення;
6) провести експериментальне дослідження запропонованого методу та порівняти його ефективність із сучасними аналогами;
7) оцінити якість результатів за критеріями непомітності, ємності та стійкості до пасивних атак.

Практичне значення отриманих результатів полягає у можливості застосування розробленого методу в системах захищеного обміну даними, цифрового водяного маркування, автентифікації мультимедійного контенту та захисту мультимедійних ресурсів у ненадійних каналах зв’язку. Поєднання адаптивного вбудовування та кодів корекції помилок забезпечує високу стійкість прихованих даних до спотворень і створює основу для подальшого вдосконалення методів стеганографії та протидії стеганалізу

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТЕГАНОГРАФІЇ ТА СТЕГОАНАЛІЗУ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1. Класифікація методів стеганографії та стегоаналізу

Стеганографія — це наука і мистецтво приховування факту передавання інформації шляхом вбудовування секретних даних у цифрові контейнери таким чином, щоб зміни залишалися непомітними для стороннього спостерігача . На відміну від криптографії, яка захищає зміст повідомлення шифруванням, стеганографія маскує саме існування комунікації, що робить її важливим інструментом у системах захисту інформації.[3].

Основними компонентами стеганографічної системи є:

1) Cover (контейнер) — вихідний цифровий об'єкт (зображення, аудіо, відео);
2) Message (повідомлення) — секретні дані, які необхідно приховати;
3) Stego-object (стегоконтейнер) — модифікований контейнер із вбудованим повідомленням;
4) Stego-key (ключ) — секретний параметр для вбудовування та вилучення даних.[1].

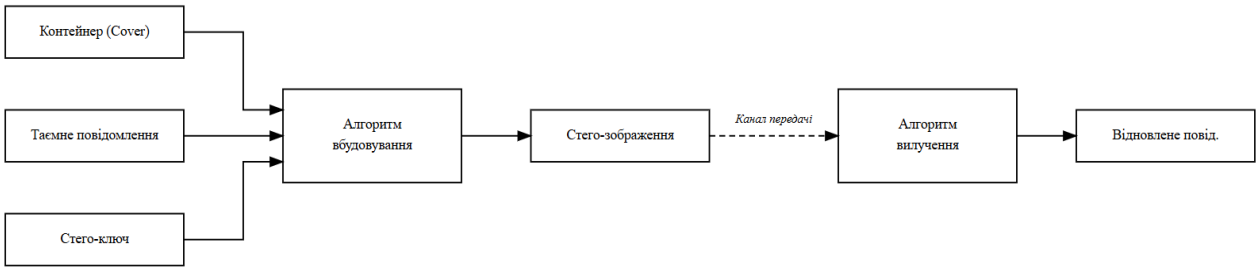

Рисунок 1.1.1 - Узагальнена схема стеганографічної системи

Сучасні стеганографічні методи класифікуються за різними критеріями.[14]. Найбільш поширеною є класифікація за доменом вбудовування інформації.

1) Методи у просторовій області (Spatial Domain)

a) LSB (Least Significant Bit) — заміна молодших бітів пікселів бітами повідомлення;
b) LSB Matching (±1) — випадкове збільшення або зменшення значення пікселя;
c) PVD (Pixel Value Differencing) — вбудовування на основі різниці сусідніх пікселів;
d) Edge-based методи — вбудовування в межові ділянки з високою текстурою.

Переваги: простота реалізації, висока ємність вбудовування.

Недоліки: низька стійкість до JPEG-стиснення, фільтрації та інших атак обробки.[7].

2) Методи у частотній області (Transform Domain)

Ці методи працюють з коефіцієнтами частотних перетворень зображення.

a) DCT-методи (Discrete Cosine Transform) - використовуються у форматі JPEG; вбудовування у середньочастотні коефіцієнти DCT-блоків 8×8; приклади: алгоритми Jsteg, F5, OutGuess .[28].
b) DWT-методи (Discrete Wavelet Transform) - багаторівнева декомпозиція на піддіапазони LL, LH, HL, HH; вбудовування у високо- або середньочастотні коефіцієнти; адаптивні стратегії на основі локальної складності зображення.
c) DFT-методи (Discrete Fourier Transform) - використання фазових або амплітудних характеристик спектра; стійкість до геометричних перетворень.

Переваги: висока стійкість до атак обробки (стиснення, фільтрація, шум).

Недоліки: нижча ємність порівняно з просторовими методами, вища обчислювальна складність.

Таблиця 1.1.1 - Порівняльна характеристика методів стеганографії.

МЕТОД	**ДОМЕН**	**ЄМНІСТЬ**	**СТІЙКІСТЬ ДО JPEG**	**СКЛАДНІСТЬ**
LSB	Просторовий	Висока	Низька	Низька
LSB Matching	Просторовий	Висока	Низька	Низька
DCT (F5)	Частотний	Середня	Висока	Середня
DWT	Частотний	Середня	Висока	Висока
DFT	Частотний	Низька	Дуже висока	Висока

3) Адаптивні методи

Сучасні підходи поєднують обидва домени та використовують адаптивну стратегію вбудовування на основі аналізу характеристик зображення.

a) HUGO (Highly Undetectable steGO) — мінімізація статистичних артефактів;
b) WOW (Wavelet Obtained Weights) — вибір коефіцієнтів на основі вейвлет-аналізу;
c) S-UNIWARD - універсальна функція спотворення для будь-якого домену.

Стегоаналіз — це сукупність методів виявлення факту приховування інформації у цифрових контейнерах.[4]. Класифікація методів стегоаналізу:

1) За типом атаки:

a) Пасивний стеганоаналіз - мета: виявити наявність прихованих даних; не спотворює стегоконтейнер; методи: статистичний аналіз, машинне навчання
b) Активний стегоаналіз - мета: зруйнувати або видалити приховані дані; спотворює зображення (стиснення, фільтрація, шум); використовується як атака на стеганосистему.

2) За методом виявлення:

a) Статистичні методи - RS-аналіз - аналіз змін регулярності/сингулярності груп пікселів; χ^2-атака — аналіз парності гістограм значень пікселів; Sample Pair Analysis — аналіз пар сусідніх пікселів.

b) Методи на основі машинного навчання - класичні ML: SVM, Random Forest на основі витягнутих ознак (SPAM, SRM); глибинне навчання: CNN-архітектури (Xu-Net, Ye-Net, SRNet, Zhu-Net); універсальні детектори: здатні виявляти різні стеганографічні алгоритми без навчання на конкретному методі.

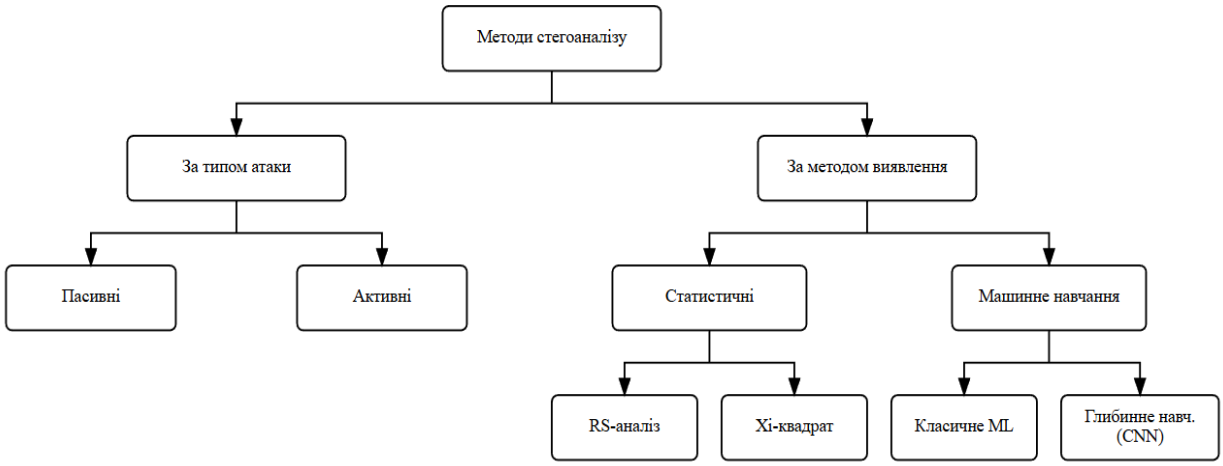

Рисунок 1.1.2 - Класифікація методів стегоаналізу

Ефективність стеганографічного методу визначається трьома основними критеріями:

a) Imperceptibility (Непомітність): відсутність візуальних або статистичних відмінностей між cover та stego; метрики: PSNR > 40 dB, SSIM > 0.95.

b) Capacity (Ємність): максимальна кількість біт секретного повідомлення, яку можна вбудувати; вимірюється у bpp (bits per pixel) або % від розміру контейнера.

c) Robustness (Стійкість): здатність зберегти приховану інформацію після атак обробки; оцінюється за BER (Bit Error Rate) після атак.

Існує фундаментальний компроміс між цими критеріями: підвищення одного призводить до зниження інших.[23].

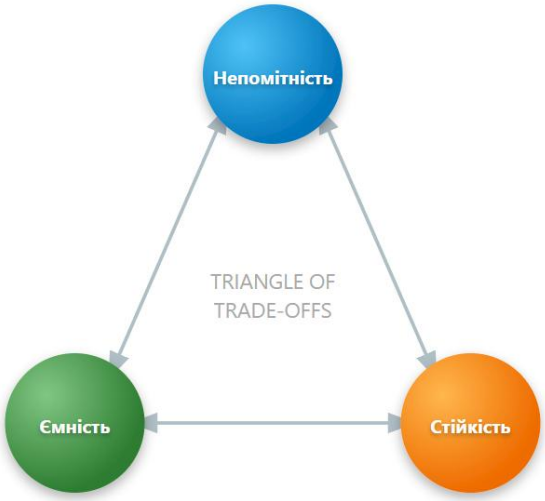

Рисунок 1.1.3 - Трикутник компромісів стеганографічної системи

1.2 Стеганографічні методи на основі дискретного вейвлет-перетворення (dwt) та використання кодів корекції помилок (ecc)

Дискретне вейвлет-перетворення (DWT) є одним із найбільш ефективних інструментів для аналізу та обробки цифрових зображень завдяки здатності розкладати сигнал на компоненти різних масштабів та частот. На відміну від дискретного косинусного перетворення (DCT), яке аналізує зображення блоками фіксованого розміру, DWT забезпечує багаторівневий аналіз всього зображення з одночасним збереженням просторової та частотної інформації.[10].

Одновимірне вейвлет-перетворення базується на розкладі сигналу за допомогою низькочастотного фільтра (Low-pass filter, L), який виділяє апроксимаційні коефіцієнти, та високочастотного фільтра (High-pass filter, H), що виділяє деталізаційні коефіцієнти. Для дискретного сигналу x[n] застосовуються операції згортки та децимації. Апроксимаційні коефіцієнти обчислюються як:

$$cA[k] = \Sigma\, x[n] \cdot h[2k - n] \qquad (1.2.1)$$

а деталізаційні:

$$cD[k] = \Sigma\, x[n] \cdot g[2k - n] \qquad (1.2.2)$$

, де імпульсні характеристики відповідних фільтрів.

Для обробки цифрових зображень використовується двовимірне DWT, яке застосовує фільтри послідовно до рядків та стовпців. Результатом одного рівня декомпозиції є чотири піддіапазони. Піддіапазон LL (Low-Low) містить апроксимацію — зменшену у 4 рази копію вихідного зображення. Піддіапазон LH (Low-High) представляє горизонтальні деталі, що відповідають вертикальним межам об'єктів. Піддіапазон HL (High-Low) містить вертикальні деталі або горизонтальні межі. Піддіапазон HH (High-High) представляє діагональні деталі та високочастотні компоненти зображення. Кожен піддіапазон містить коефіцієнти розміром N/2 × M/2, де N×M — розмір вихідного зображення.[20].

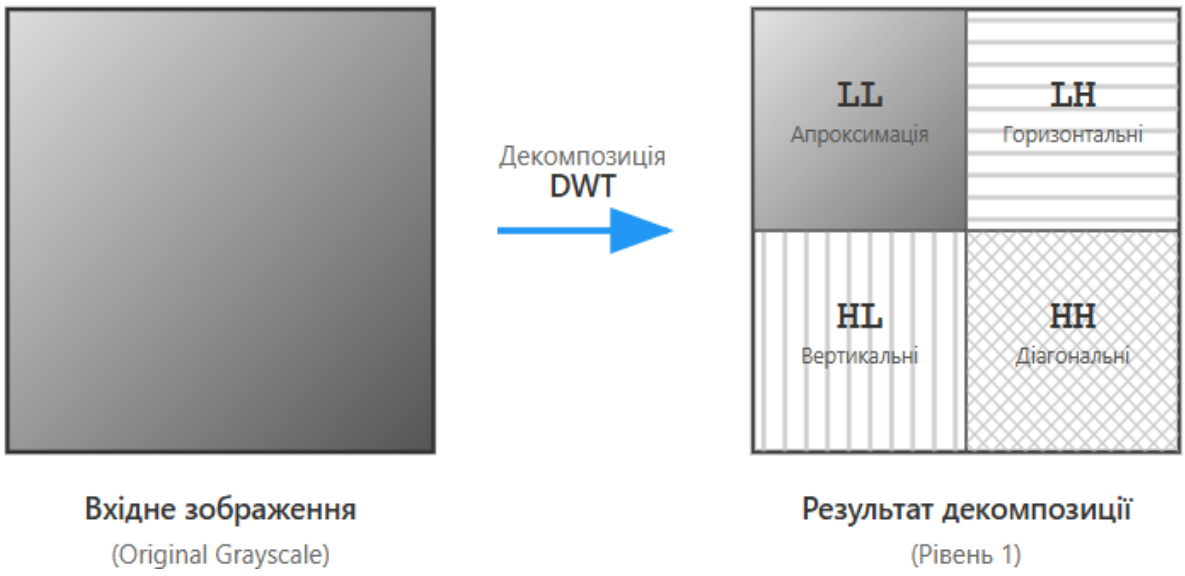

Рисунок 1.2.1 - Схема одного рівня 2D-DWT декомпозиції зображення

DWT підтримує ієрархічну декомпозицію, при якій піддіапазон LL може бути повторно розкладений на наступному рівні . Для зображення розміром 512×512 пікселів на першому рівні отримуємо чотири піддіапазони

розміром 256×256 кожен. На другому рівні маємо вже сім піддіапазонів, де LL_2 має розмір 128×128, а піддіапазони LH_2, HL_2, HH_2 також 128×128, при цьому зберігаються піддіапазони першого рівня LH_1, HL_1, HH_1 розміром 256×256. Третій рівень декомпозиції дає загалом десять піддіапазонів різних розмірів.

Вибір оптимального рівня декомпозиції для стеганографічного вбудовування є фундаментальною архітектурною проблемою, яка визначається складним компромісом між трьома взаємовиключними вимогами: стійкістю до зовнішніх впливів, інформаційною ємністю контейнера та візуальною непомітністю прихованих даних. Цей баланс особливо критичний при використанні частотних перетворень, де кожен рівень розкладання має унікальні енергетичні та статистичні властивості.

Вищі рівні декомпозиції, такі як другий або третій, демонструють значно кращу стійкість до втратного стиснення, наприклад, стандарту JPEG, а також до фільтрації та додавання шуму.[3]. Це пояснюється тим, що алгоритми стиснення в першу чергу видаляють високочастотні деталі, які знаходяться на першому рівні, тоді як на глибших рівнях відбувається узагальнення деталей та концентрація основної енергії зображення. Інформація, вбудована в ці низько- та середньочастотні піддіапазони, стає частиною значущої структури зображення, що дозволяє їй пережити процес квантування при стисненні.

Водночас виникає дилема між ємністю та непомітністю: хоча математична природа декомпозиції забезпечує нижчим рівням (особливо першому) максимальну кількість коефіцієнтів для вбудовування великих обсягів даних, це робить їх вразливими, тоді як перехід на рівні 2–3, попри зниження ємності, дозволяє краще адаптувати зміни до особливостей зорової системи людини. Модифікація цих більш узагальнених характеристик при зворотному перетворенні розподіляється по площині зображення, що дозволяє уникнути різкого контрасту та «зернистості», характерних для першого рівня, роблячи другий рівень своєрідною «золотою серединою» для

збереження балансу між достатнім обсягом повідомлення та відсутністю візуальних артефактів.[20].

Рисунок 1.2.2 - Трирівнева DWT-декомпозиція зображення Lena

Вибір конкретного піддіапазону для вбудовування суттєво впливає на властивості стеганографічного методу. Піддіапазон LL характеризується максимальною стійкістю до різних типів атак, проте будь-які модифікації в ньому візуально помітні, оскільки цей піддіапазон містить основний зміст зображення, що також створює високий ризик виявлення методами стегоаналізу. Піддіапазони LH та HL, які представляють середні частоти, забезпечують розумний баланс між непомітністю та стійкістю і тому часто

використовуються у практичних стеганографічних методах, хоча вони залишаються частково чутливими до JPEG-стиснення.

Піддіапазон HH, що містить високочастотні компоненти, має особливі властивості для стеганографічного застосування . Високочастотні деталі найменш помітні для людського зору через особливості системи візуального сприйняття, а зміни в HH-коефіцієнтах практично не впливають на візуальне сприйняття зображення. Основним недоліком цього піддіапазону є висока чутливість до атак фільтрації та стиснення. Проте саме ця особливість робить HH-коефіцієнти ідеальним вибором для методів, які використовують коди корекції помилок, здатні компенсувати спотворення від атак . Експериментальні дослідження показують, що вбудовування в HH-коефіцієнти другого рівня декомпозиції забезпечує показник PSNR понад 42 dB при ємності до 0.1 біт на піксель.

Таблиця 1.2.1 - Порівняння характеристик DWT-піддіапазонів для стеганографії

ПІДДІАПАЗОН	**ВІЗУАЛЬНА ВАЖЛИВІСТЬ**	**СТІЙКІСТЬ ДО JPEG**	**НЕПОМІТНІСТЬ ЗМІН**	**РЕКОМЕНДАЦІЯ**
LL	Дуже висока	Максимальна	Низька	Не використовувати
LH	Середня	Висока	Середня	Рекомендовано
HL	Середня	Висока	Середня	Рекомендовано
HH	Низька	Низька	Дуже висока	З ECC

Для модифікації вейвлет-коефіцієнтів з метою приховування інформації застосовуються різні підходи. Найпростішим є метод прямої заміни молодших бітів коефіцієнтів, який реалізується за формулою:

$$cD[k] = \Sigma\, x[n] \cdot g[2k - n] \quad (1.2.3)$$

де c[i,j] позначає оригінальний коефіцієнт, а m[k] — біт повідомлення, що вбудовується . Цей метод характеризується низькою стійкістю та легко виявляється статистичними методами аналізу.

Більш досконалим є метод квантування коефіцієнтів, при якому вбудовування біта здійснюється шляхом округлення коефіцієнта до парного або непарного значення.[29]. Якщо необхідно вбудувати біт зі значенням 0, коефіцієнт округлюється як:

$$c'[i,j] = 2 \cdot round(c[i,j] / (2 \cdot \Delta)) \quad (1.2.4)$$

, а для біта зі значенням 1 застосовується формула:

$$c'[i,j] = 2 \cdot round(c[i,j] / (2 \cdot \Delta)) + \Delta \quad (1.2.5)$$

, де параметр Δ визначає крок квантування та силу вбудовування.

Сучасний адаптивний підхід S-UNIWARD використовує функцію спотворення для мінімізації ймовірності виявлення стеганографічного вбудовування. Функція спотворення обчислюється як:

$$D(c, c') = \Sigma\, |W(c[i,j] - c'[i,j])| \quad (1.2.6)$$

де W представляє вейвлет-фільтр, що оцінює локальну складність зображення. Згідно з цим підходом вбудовування відбувається переважно в текстурні області з високою складністю, що забезпечує мінімальні статистичні артефакти.

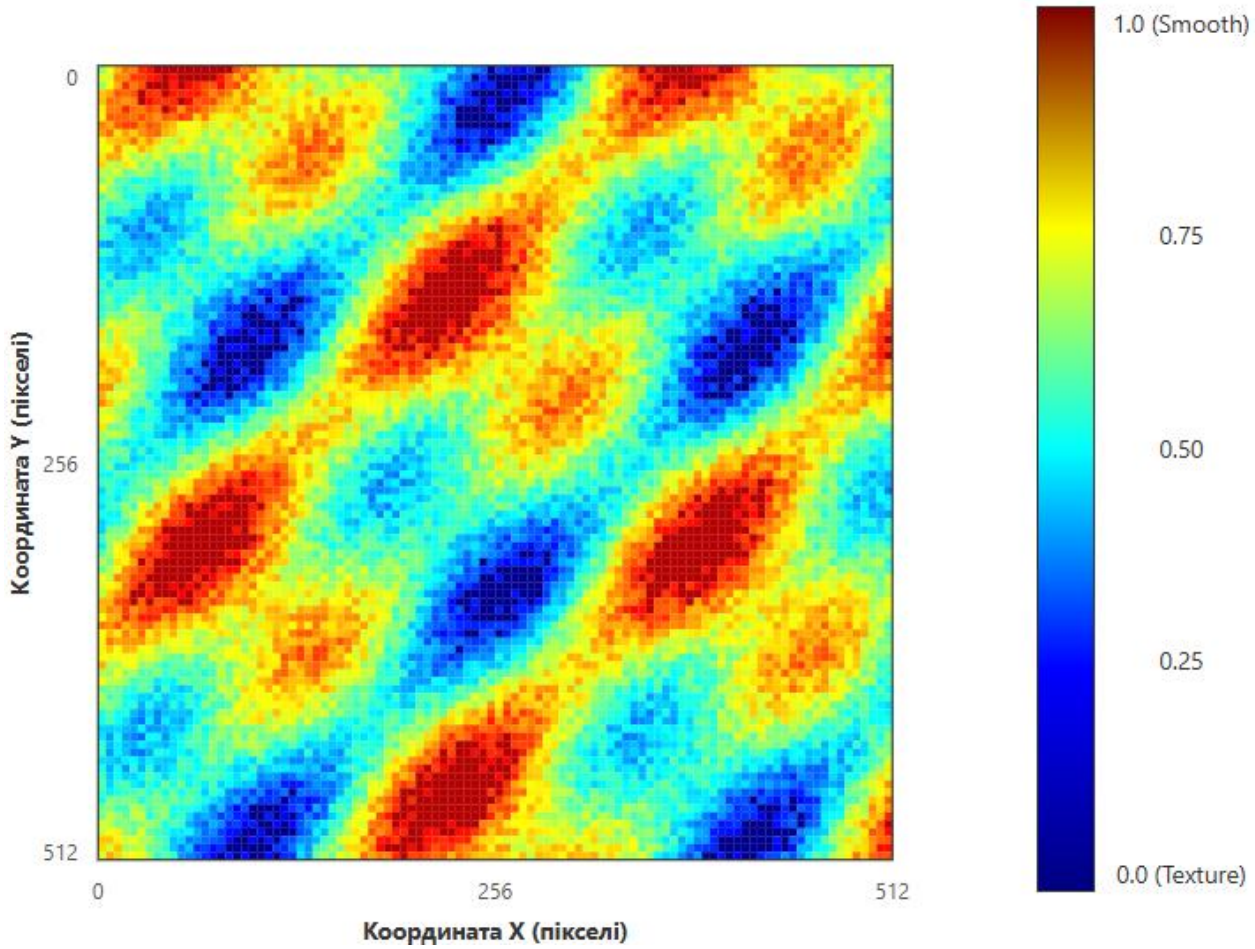

Рисунок 1.2.2 - Приклад адаптивної карти вбудовування S-UNIWARD

Пасивні атаки, такі як JPEG-стиснення, фільтрація та додавання шуму, спричиняють спотворення вбудованих даних, особливо у високочастотних коефіцієнтах. Коди корекції помилок (ECC, Error Correction Codes) забезпечують можливість відновлення повідомлення навіть за наявності помилок у витягнутих даних

Принцип роботи кодів корекції базується на додаванні надмірності до оригінального повідомлення . Якщо повідомлення має довжину k біт, то після кодування воно перетворюється на n біт, де $n > k$. Різниця $r = n - k$ представляє надмірність, яка забезпечує можливість виправлення до t помилок, де конкретне значення t залежить від типу та параметрів використовуваного коду.

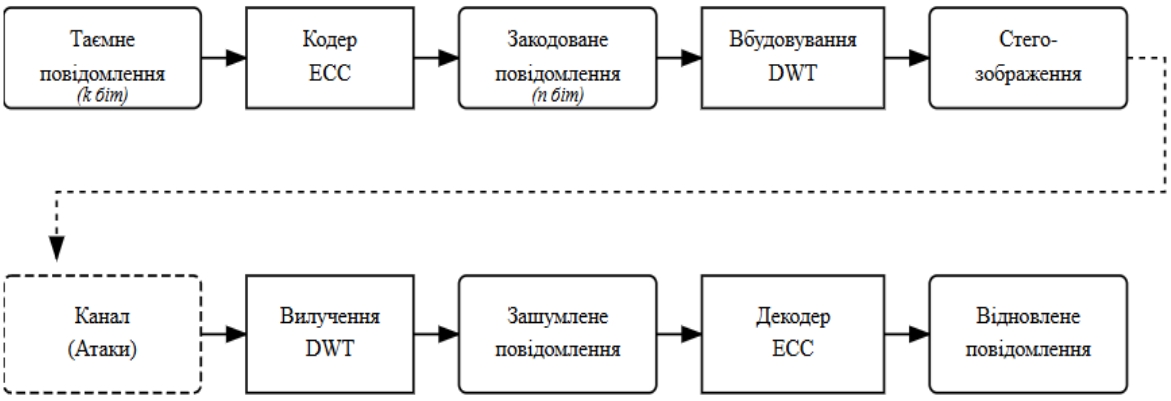

Рисунок 1.2.3 - Схема застосування ECC у стеганографічній системі

BCH-коди (Bose-Chaudhuri-Hocquenghem) є блоковими циклічними кодами, які широко застосовуються у стеганографічних системах. Код BCH характеризується параметрами (n, k, t), де n означає довжину кодового слова, k - довжину інформаційного блоку, а t - кількість помилок, які код здатен виправити. Наприклад, код BCH(127, 64, 10) кодує 64 біти інформації у 127-бітне кодове слово та може виправити до 10 помилок. Перевагами BCH-кодів є ефективне виправлення випадкових помилок та наявність добре розробленої теорії з ефективними алгоритмами декодування. Серед недоліків варто відзначити фіксованість параметрів та меншу ефективність при обробці серій послідовних помилок.[14].

Reed-Solomon коди є найпопулярнішими у застосуваннях стеганографії та цифрових водяних знаків. На відміну від BCH, які працюють з бітами, RS-коди оперують символами, зазвичай 8-бітними байтами. Код RS(n, k) здатен виправити до (n-k)/2 пошкоджених символів . Приклад коду RS(255, 223) кодує 223 байти у 255-байтове слово та може виправити до 16 пошкоджених байтів. Reed-Solomon коди відмінно справляються з серіями послідовних помилок, мають максимальну кодову відстань для заданої надмірності та існують у вигляді ефективних апаратних і програмних реалізацій. Основним недоліком є вища обчислювальна складність порівняно з BCH-кодами.[14].

LDPC-коди (Low-Density Parity-Check) представляють сучасний клас кодів корекції з продуктивністю, близькою до теоретичної межі Шеннона. і коди використовують ітеративне декодування на основі алгоритму Belief Propagation та забезпечують гнучкість у виборі довжини і швидкості кодування. LDPC-коди застосовуються у сучасних стандартах бездротового зв'язку, таких як Wi-Fi (802.11n), DVB-S2 та 5G. Їхньою перевагою є найкраща продуктивність при високих рівнях шуму та гнучкість налаштування параметрів. Проте складність реалізації та висока обчислювальна вартість декодування обмежують їх застосування у стеганографії переважно дослідницькими роботами.[14].

Таблиця 1.2.2 - Порівняння кодів корекції помилок для стеганографії

КОД	**ТИП ПОМИЛОК**	**СКЛАДНІСТЬ**	**ЕФЕКТИВНІСТЬ**	**ЗАСТОСУВАННЯ В СТЕГАНОГРАФІЇ**
BCH	Випадкові	Середня	Середня	Часто
Reed-Solomon	Burst errors	Середня	Висока	Дуже часто
LDPC	Універсальні	Висока	Максимальна	Рідко

Комбінування дискретного вейвлет-перетворення з кодами корекції помилок забезпечує потужний синергетичний ефект, створюючи дворівневий захист даних, де фізична стійкість методу вбудовування компенсується математичною здатністю кодів відновлювати втрачену інформацію. Схема вбудовування реалізується наступним чином: на

першому етапі секретне повідомлення Mдовжиною kбіт проходить процедуру ECC-кодування, що додає необхідну надлишковість для боротьби з шумом; у результаті формується розширена послідовність $M_{encoded}$ довжиною nбіт $(n > k)$. Паралельно виконується DWT-декомпозиція зображення-контейнера, зазвичай до другого рівня, що дозволяє ефективно розділити низькочастотну та високочастотну складові. Щоб мінімізувати візуальні артефакти, на основі розрахунку критерію текстурності (наприклад, локальної дисперсії) відбувається адаптивний відбір коефіцієнтів піддіапазону HH_2, які відповідають за діагональні деталі та є найменш чутливими для людського ока. Закодоване повідомлення $M_{encoded}$ вбудовується у ці вибрані області, після чого застосовується зворотне вейвлет-перетворення для синтезу фінального стегозображення.

Процес витягування інформації здійснюється у зворотному порядку та враховує ймовірність спотворення даних у каналі передачі. Стегозображення, яке може бути модифікованим внаслідок атак типу JPEG-стиснення, фільтрації або шуму, піддається DWT-декомпозиції з тими ж параметрами, що й при вбудовуванні. З визначених позицій HH_2-коефіцієнтів зчитується потенційно пошкоджена послідовність $M_{extracted}$, яка може містити інвертовані біти. На цьому етапі критичну роль відіграє ECC-декодер, який обробляє $M_{extracted}$, локалізуючи та виправляючи помилки на основі перевірочних бітів. У результаті формується скоригована версія повідомлення $M_{corrected}$, і за умови, що щільність помилок не перевищує теоретичної межі корекційної здатності обраного коду, система гарантує отримання на виході даних, повністю ідентичних оригінальному секретному повідомленню M.

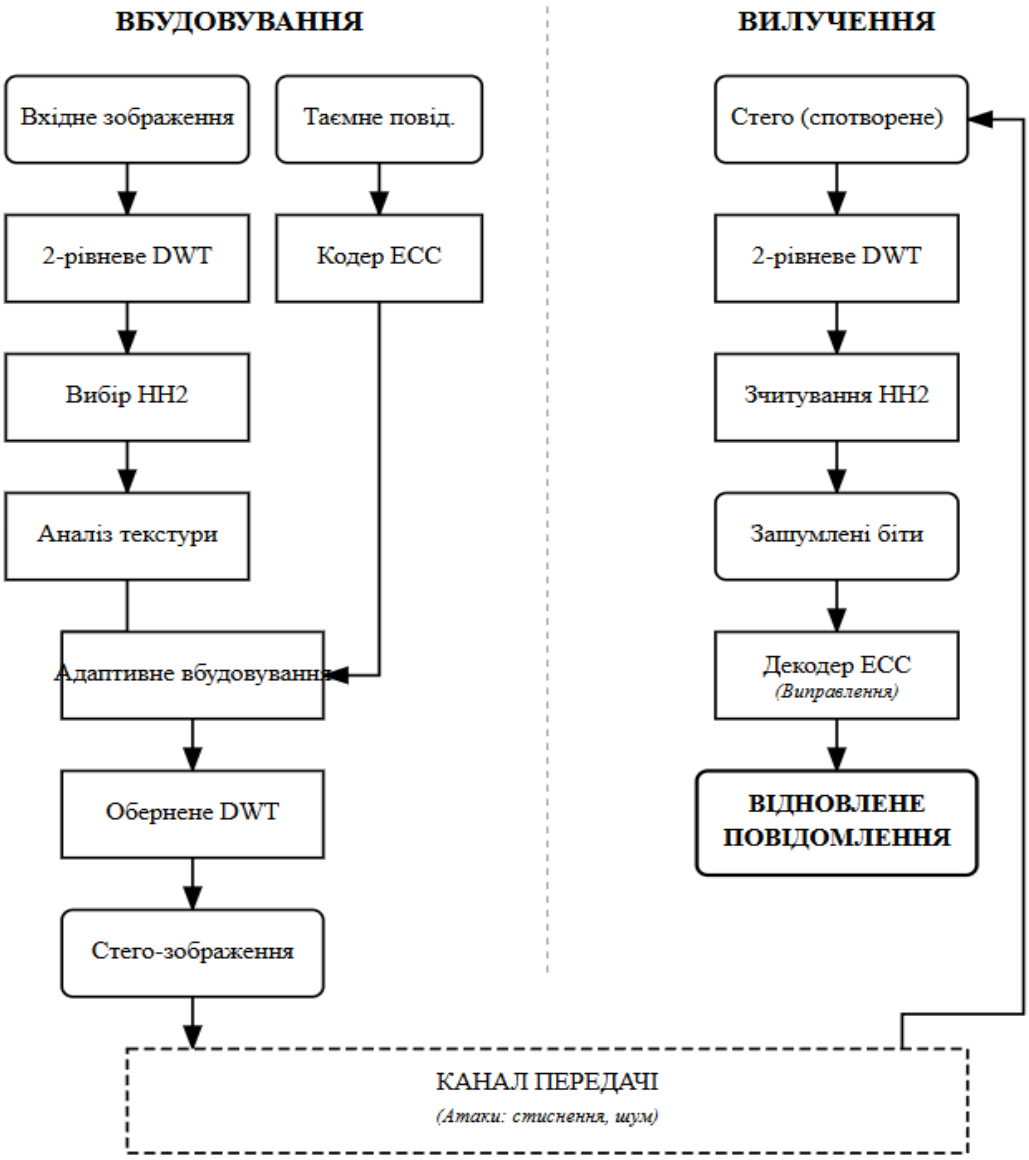

Рисунок 1.2.4 - Детальна блок-схема методу DWT+ECC

Використання кодів корекції помилок неминуче зменшує ефективну ємність контейнера через необхідність додавання надмірної інформації. Ефективність використання ємності характеризується коефіцієнтом кодування R = k/n, який показує частку корисної інформації у закодованому повідомленні.

Розглянемо конкретний приклад для зображення розміром 512×512 пікселів. HH_2-піддіапазон другого рівня декомпозиції може вмістити приблизно 16384 біти інформації. При використанні коду Reed-Solomon RS(255, 223) коефіцієнт кодування становить R = 223/255 ≈ 0.875, що означає

ефективну корисну ємність близько 14336 біт. Таким чином, приблизно 2048 біт використовуються як надмірність для забезпечення корекції помилок.

Вибір конкретних параметрів коду корекції визначається очікуваним рівнем атак та вимогами до стійкості системи . У випадку сильних атак, таких як JPEG-стиснення з якістю Q=50 або додавання інтенсивного шуму, необхідна висока надмірність, наприклад код RS(255, 191) з коефіцієнтом R = 0.75. При такому виборі частота бітових помилок після атаки може становити 15-20%, але після застосування декодування вона знижується до нуля. Для помірних атак, як-от JPEG з якістю Q=75 або слабкий шум, достатньою є помірна надмірність коду RS(255, 223) з R = 0.875, що дозволяє виправити початкову частоту помилок 5-8%. У випадку слабких атак, таких як JPEG високої якості (Q=90), можна використовувати мінімальну надмірність коду RS(255, 239) з R = 0.938, який справляється з початковою частотою помилок 1-2%.

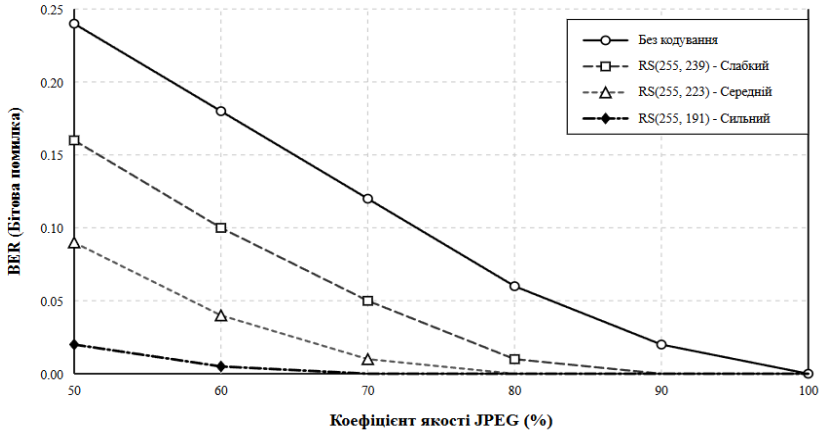

Рисунок 1.2.5 - Залежність BER від параметрів RS-коду при різних рівнях JPEG-стиснення

1.3. Методи стеганографії у просторовій області та їхні обмеження

Методи стеганографії у просторовій області характеризуються прямою модифікацією значень пікселів зображення-контейнера без попереднього застосування частотних перетворень. Основною перевагою таких методів є простота реалізації та висока обчислювальна ефективність, що робить їх

привабливими для застосувань, де швидкість обробки має критичне значення. Водночас просторові методи демонструють суттєві обмеження щодо стійкості до поширених атак обробки зображень, що обмежує їхнє практичне застосування у сценаріях з високими вимогами до надійності передавання прихованої інформації.

Метод LSB (Least Significant Bit) - метод заміни молодшого значущого біта є найбільш відомим та історично першим підходом до стеганографії у цифрових зображеннях. Ідея методу базується на властивості людського зору, яке не здатне розрізнити зміну інтенсивності пікселя на одиницю у 256-градаціях відтінків сірого або кольорових компонент. Для 8-бітного представлення пікселя зміна молодшого біта змінює значення інтенсивності максимум на одиницю, що залишається непомітним візуально.

Процес вбудовування реалізується шляхом заміни молодшого біта кожного пікселя послідовними бітами секретного повідомлення.[5]. Математично це можна записати як

$$p'[i,j] = (p[i,j] \,\&\, 0xFE) \mid m[k] \qquad (1.3.1)$$

, де p[i,j] представляє значення оригінального пікселя у позиції (i,j), m[k] — k-тий біт повідомлення, а операція & 0xFE обнуляє молодший біт пікселя перед вбудовуванням нового значення. Для кольорових зображень у форматі RGB метод може застосовуватися до всіх трьох каналів або вибірково до певних компонент, що впливає на ємність та непомітність вбудовування.

Теоретична ємність методу LSB для зображення розміром M×N пікселів становить M×N біт для монохромних зображень та 3×M×N біт для RGB-зображень . Наприклад, зображення розміром 512×512 пікселів може вмістити до 262144 біт або 32 кілобайти секретної інформації без використання стиснення. Така висока ємність є однією з головних переваг методу LSB порівняно з частотними підходами.

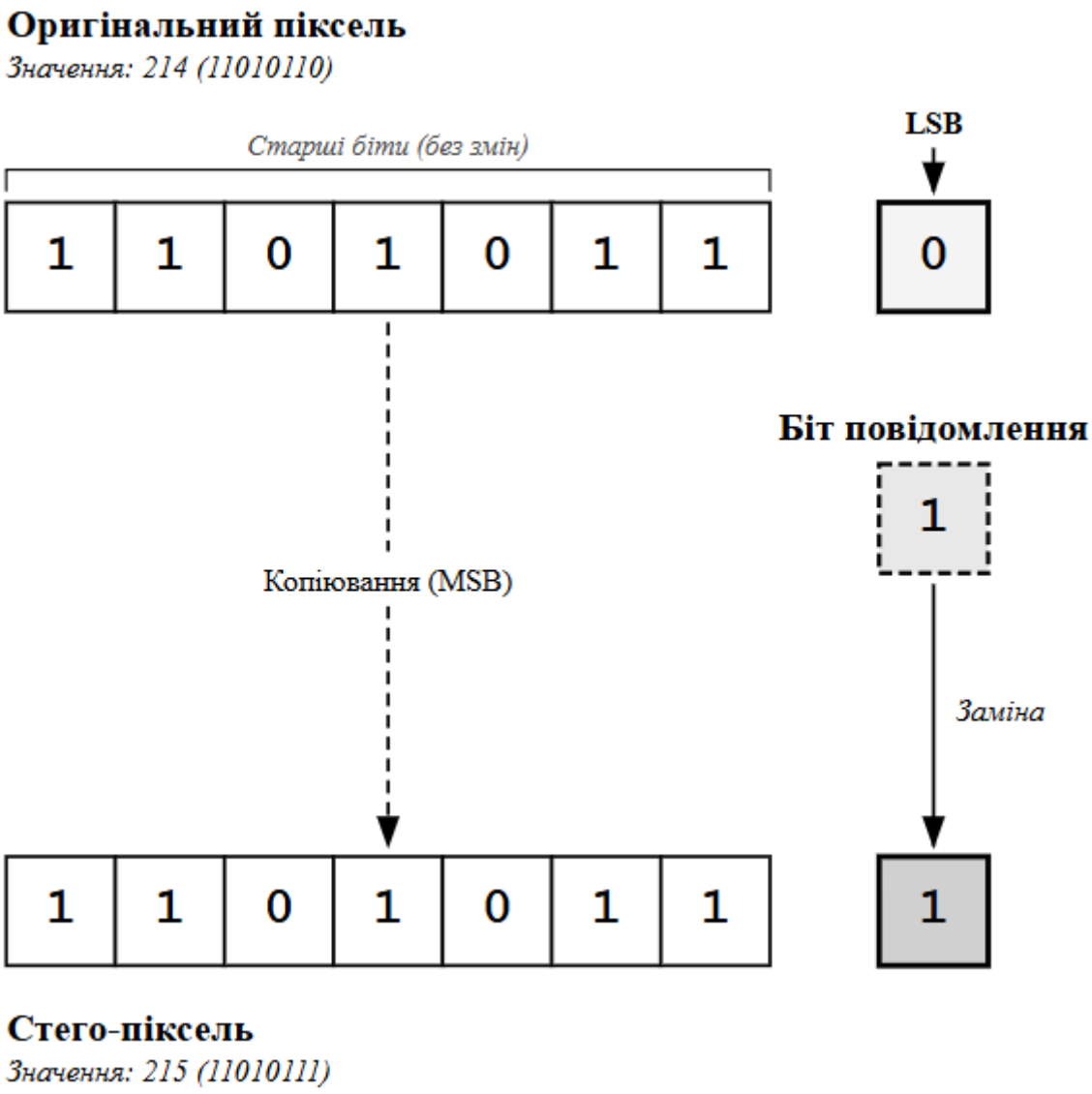

Рисунок 1.3.1 - Схема LSB-вбудовування у піксель зображення

Метод LSB Matching, також відомий як ±1-вбудовування, був розроблений як удосконалення класичного LSB з метою протидії статистичним методам стегоаналізу. На відміну від прямої заміни молодшого біта, цей метод модифікує значення пікселя випадковим чином із збереженням парності або непарності залежно від біта повідомлення.[6].

Алгоритм вбудовування працює наступним чином. Якщо молодший біт пікселя вже збігається зі значенням біта повідомлення, піксель залишається незмінним. У протилежному випадку значення пікселя випадково збільшується або зменшується на одиницю. Формально це записується як:

$$\text{якщо } LSB(p[i,j]) = m[k], \text{то } p'[i,j] = p[i,j] \qquad (1.3.2)$$

,інакше $$p'[i,j] = p[i,j] + rand\{-1,+1\} \quad (1.3.3)$$

, де функція rand{-1, +1} генерує випадково -1 або +1 з рівною ймовірністю. Важливо врахувати граничні випадки: якщо p[i,j] = 0, можливе лише збільшення, а якщо p[i,j] = 255, можливе лише зменшення.

Ключова відмінність LSB Matching від класичного LSB полягає у збереженні статистичних властивостей зображення . Класичний LSB створює характерну аномалію у розподілі значень пікселів — пари сусідніх значень (2k, 2k+1) стають аномально частими. LSB Matching уникає цього ефекту через випадкову модифікацію у обох напрямках, що робить його більш стійким до χ^2-атаки та аналізу пар значень (Pairs of Values analysis)

Розвиток методів статистичного стегоаналізу стимулював створення адаптивних підходів, які вбудовують інформацію не рівномірно по всьому зображенню, а вибірково у певні ділянки з урахуванням локальних характеристик.

Метод PVD (Pixel Value Differencing) базується на різниці значень сусідніх пікселів. Зображення розділяється на блоки по два сусідні пікселі, і для кожного блоку обчислюється різниця за формулою:

$$d = |p[i] - p[i+1]| \quad (1.3.4)$$

Кількість бітів, що вбудовуються в блок, визначається величиною різниці: більша різниця дозволяє вбудувати більше бітів без помітних артефактів. Це пояснюється тим, що в областях з високою текстурою (великі різниці) зміни менш помітні порівняно з гладкими областями. Типово діапазон різниць розбивається на кілька піддіапазонів, кожному з яких відповідає певна ємність вбудовування від 1 до 5 біт на блок.

Edge-based методи (методи на основі меж) концентрують вбудовування у ділянках зображення з високим градієнтом, які відповідають межам об'єктів . Для кожного пікселя обчислюється локальний градієнт за допомогою операторів Sobel, Canny або інших детекторів меж. Вбудовування відбувається переважно у пікселях з високим значенням градієнта, оскільки людський зір менш чутливий до змін у таких областях порівняно з гладкими

фрагментами зображення . Така стратегія дозволяє досягти кращого співвідношення між ємністю та непомітністю.

Метод HUGO (Highly Undetectable steGO) представляє сучасний адаптивний підхід, який мінімізує статистичні артефакти у високовимірному просторі ознак . Метод використовує функцію вартості (cost function), яка оцінює вплив модифікації кожного пікселя на сукупність статистичних характеристик зображення. Вбудовування виконується переважно у пікселі з низькою вартістю модифікації, що мінімізує ймовірність виявлення сучасними стегоаналітичними детекторами на основі машинного навчання.

Найсуттєвішим обмеженням просторових методів є катастрофічна втрата вбудованої інформації при JPEG-стисненні. JPEG використовує дискретне косинусне перетворення та квантування коефіцієнтів, що призводить до втрати молодших бітів пікселів навіть при високій якості стиснення.[16].

Експериментальні дослідження показують наступну динаміку руйнування LSB-повідомлень. При JPEG-стисненні з якістю 90% частота бітових помилок (BER) становить приблизно 30-40%, що робить повідомлення нечитабельним без застосування кодів корекції. При якості 80% BER зростає до 45-55%, а при якості 70% перевищує 60%. Навіть максимальна якість JPEG (95-98%) спричиняє BER на рівні 10-20%, що критично для більшості застосувань

Порівняльний аналіз стійкості різних методів показує, що LSB Matching демонструє дещо кращу стійкість порівняно з класичним LSB завдяки меншій кореляції між сусідніми пікселями. Адаптивні методи типу PVD виявляють стійкість на рівні LSB Matching. Проте всі просторові методи поступаються частотним підходам на основі DWT або DCT, які забезпечують BER менше 5% навіть при JPEG якості 70% за умови використання кодів корекції помилок.

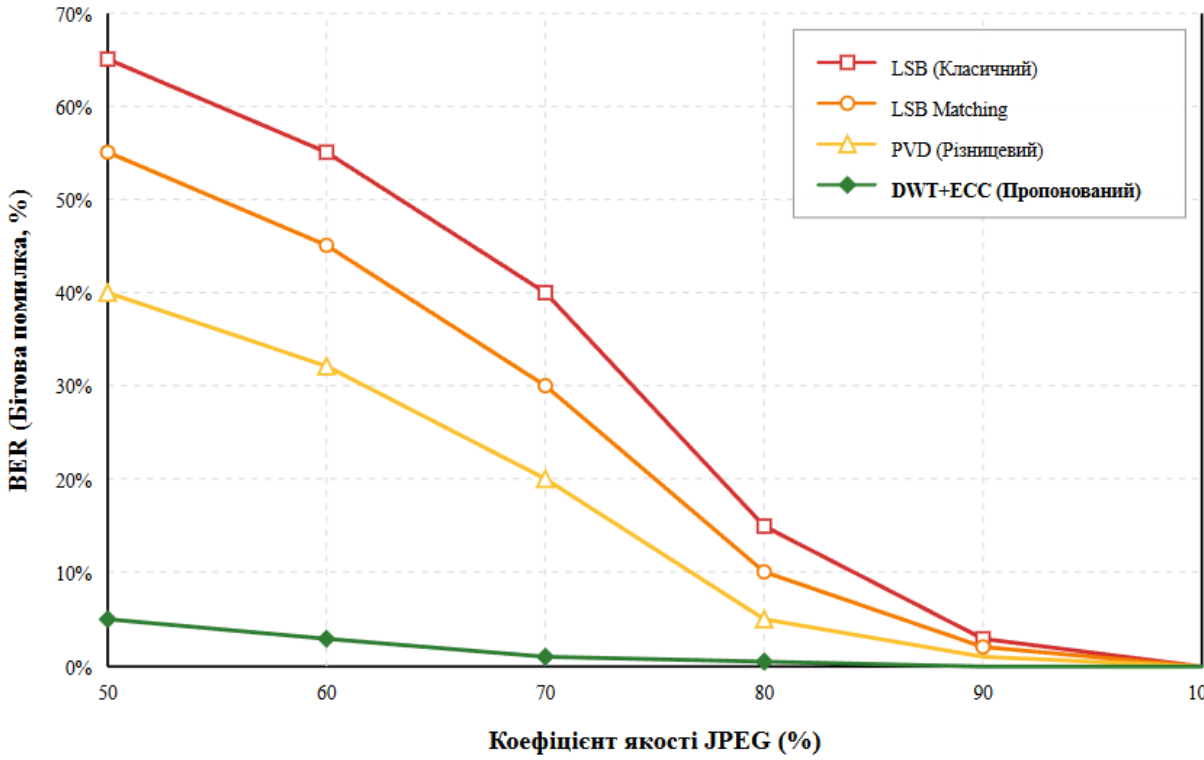

Рисунок 1.3.2 - Залежність BER від якості JPEG-стиснення для різних методів

Просторові методи стеганографії створюють характерні статистичні артефакти, які можуть бути виявлені методами стегоаналізу навіть без знання конкретного алгоритму вбудовування.

RS-аналіз (Regular-Singular analysis) є одним із найефективніших методів виявлення LSB-стеганографії . Метод базується на аналізі змін статистичних властивостей груп пікселів при застосуванні спеціальних функцій дискримінації. Зображення розділяється на групи по кілька пікселів, і для кожної групи визначається, чи є вона регулярною (R), сингулярною (S) чи незмінною (U) . LSB-вбудовування порушує природний баланс між кількістю R- та S-груп, що дозволяє не лише виявити факт вбудовування, але й оцінити відсоток заповнення контейнера. Точність RS-аналізу для класичного LSB перевищує 95% при заповненні понад 10% ємності.

χ^2-атака (Chi-square attack) аналізує парність розподілу значень пікселів . Природні зображення мають певний розподіл частот значень 2k та 2k+1 для кожного k. LSB-вбудовування вирівнює ці частоти, оскільки заміна молодшого біта перетворює значення 2k у 2k+1 і навпаки з приблизно однаковою ймовірністю. Статистика χ^2 порівнює спостережуваний розподіл із очікуваним для стегозображення, що дозволяє виявити LSB-вбудовування з високою точністю навіть при малій ємності заповнення.[18].

Аналіз гістограм виявляє аномалії у розподілі інтенсивностей пікселів . LSB-вбудовування призводить до згладжування гістограми через вирівнювання частот парних та непарних значень. Візуальний аналіз гістограм часто дозволяє експерту ідентифікувати стегозображення навіть без формальних статистичних тестів. Sample Pairs Analysis розглядає пари сусідніх пікселів і аналізує зміни у їх статистичних властивостях, що також ефективно для виявлення як LSB, так і LSB Matching.

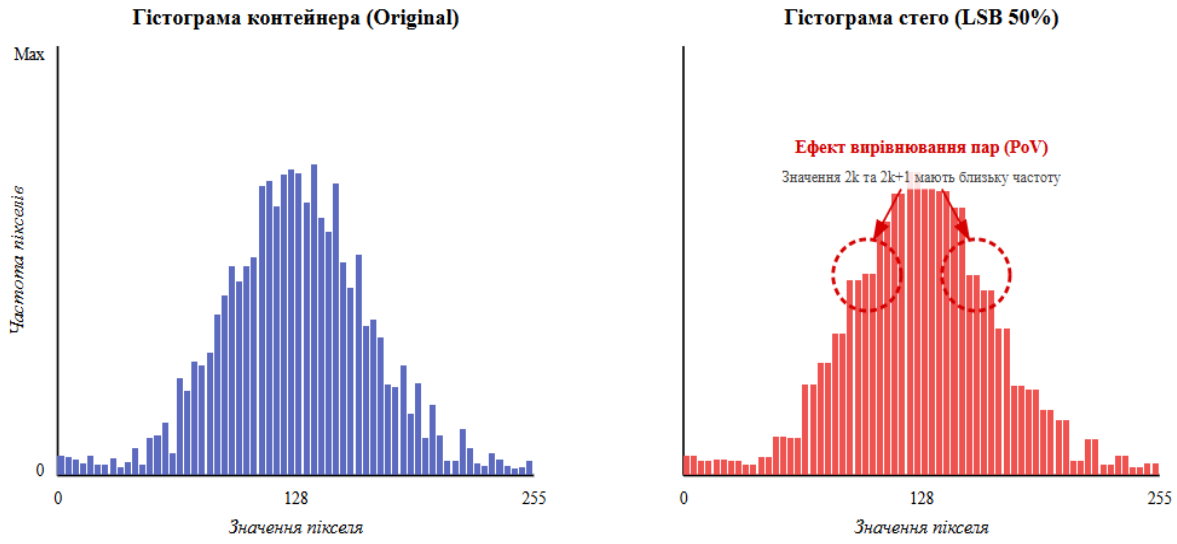

Рисунок 1.3.3 - Гістограми зображення до та після LSB-вбудовування

Незважаючи на покращення стійкості до статистичного стегоаналізу, адаптивні просторові методи все ще демонструють фундаментальні обмеження. По-перше, концентрація вбудовування у текстурних областях

або межах створює нові типи артефактів, які можуть бути виявлені спеціалізованими детекторами.

По-друге, адаптивні методи не вирішують проблему вразливості до JPEG-стиснення, оскільки втрата молодших бітів відбувається незалежно від стратегії їх вибору.[17].

Сучасні методи стегоаналізу на основі глибинного навчання, такі як SRNet або Xu-Net, демонструють високу ефективність виявлення навіть адаптивних просторових методів. Згорткові нейронні мережі навчаються виявляти складні статистичні закономірності у високовимірному просторі ознак, які неможливо усунути, без переходу, до частотної області. Експериментальні результати показують, що точність виявлення HUGO або S-UNIWARD у просторовій області досягає 75-85% при використанні CNN-детекторів, тоді як для частотних методів з адаптивним вбудовуванням точність знижується до 55-65%.[6].

Таблиця 1.3.1 - Порівняльна характеристика методів просторової стеганографії

МЕТОД	**ЄМНІСТЬ**	**СТІЙКІСТЬ ДО JPEG**	**СТІЙКІСТЬ ДО RS-АНАЛІЗУ**	**СКЛАДНІСТЬ**	**ПРАКТИЧНЕ ЗАСТОСУВАННЯ**
LSB	Дуже висока	Дуже низька	Дуже низька	Мінімальна	Обмежене
LSB Matching	Висока	Дуже низька	Низька	Низька	Обмежене
PVD	Середня	Низька	Середня	Середня	Помірне

Продовження таблиці 1.3.1 – Порівняльна характеристика методів просторової стеганографії

Edge-based	Середня	Низька	Середня	Середня	Помірне
HUGO	Середня	Низька	Висока	Висока	Дослідження

1.4. Пасивні та активні атаки на стеганографічні методи

Стеганографічні системи у реальних умовах функціонування стикаються з різноманітними впливами, які можуть спотворити або повністю знищити приховану інформацію. За характером дії та цілями такі впливи класифікуються на пасивні та активні атаки, кожна з яких має специфічні характеристики та потребує різних підходів до протидії.

Пасивні атаки спрямовані виключно на виявлення факту наявності прихованої інформації у контейнері без спроб її знищення або модифікації. Такі атаки базуються на статистичному аналізі властивостей зображення та пошуку аномалій, які відрізняють стегозображення від звичайного. Типовими представниками пасивних атак є RS-аналіз, χ^2-тест, аналіз гістограм, методи на основі машинного навчання та глибинні нейронні мережі для стегоаналізу. Метою зловмисника у цьому випадку є ідентифікація підозрілих зображень для подальшого блокування комунікаційного каналу або застосування активних атак.

Активні атаки, навпаки, мають на меті пошкодження або повне знищення прихованої інформації через навмисну або ненавмисну модифікацію стегоконтейнера. Важливою особливістю активних атак є те, що вони часто виникають не як цілеспрямовані дії зловмисника, а як побічний ефект звичайної обробки зображень у процесі передавання, зберігання або публікації. До активних атак відносяться JPEG-стиснення, додавання шуму, фільтрація, геометричні перетворення, обрізання та зміна розміру зображення.

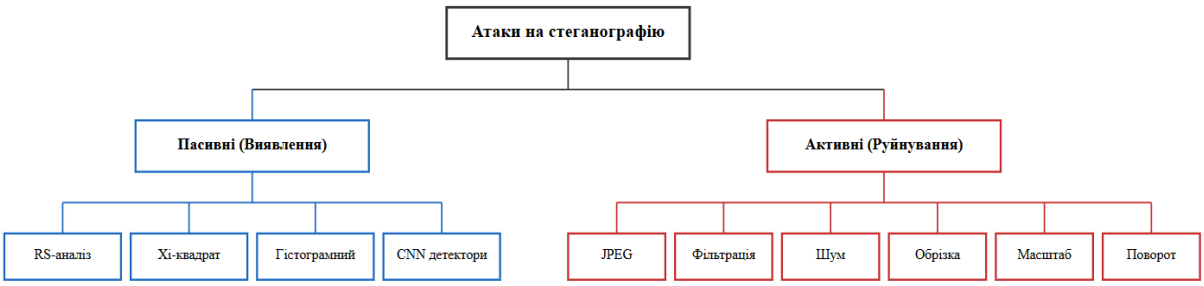

Рисунок 1.4.1 - Класифікація атак на стеганографічні системи

JPEG-стиснення є найпоширенішою та найбільш руйнівною атакою для методів просторової стеганографії. Алгоритм JPEG виконує послідовність операцій, кожна з яких вносить спотворення у вбудовані дані. Спочатку зображення конвертується з RGB у колірний простір YCbCr, де компонента Y представляє яскравість, а Cb та Cr — кольорову інформацію. Потім зображення розбивається на блоки розміром 8×8 пікселів, до кожного з яких застосовується дискретне косинусне перетворення.[11].

Найбільш руйнівною стадією JPEG є квантування DCT-коефіцієнтів, при якому кожен коефіцієнт ділиться на відповідне значення з таблиці квантування та округлюється до найближчого цілого числа. Таблиця квантування визначається фактором якості Q (зазвичай від 1 до 100), де більші значення Q відповідають кращій якості зображення та меншій компресії. Високочастотні DCT-коефіцієнти квантуються з більшими кроками, що призводить до їх обнулення або суттєвого спотворення. Саме ця властивість робить JPEG критично небезпечним для LSB-методів, оскільки молодші біти пікселів змінюються непередбачувано.

Вплив JPEG-стиснення на різні стеганографічні методи суттєво відрізняється. Для LSB-методів навіть високоякісне стиснення з Q=90 призводить до частоти бітових помилок 25-35%, що робить повідомлення нечитабельним. При зниженні якості до Q=80 BER зростає до 40-50%, а при Q=70 перевищує 55-60%. DCT-методи, які вбудовують інформацію безпосередньо у DCT-коефіцієнти середніх частот, демонструють значно кращу стійкість з BER близько 5-10% при Q=70. DWT-методи з

вбудовуванням у коефіцієнти другого рівня декомпозиції показують BER у межах 8-15% при Q=70, що може бути скориговано кодами Reed-Solomon.

Експериментальні дослідження показують, що повторне JPEG-стиснення (зображення спочатку стискається, потім розпаковується і стискається знову) призводить до накопичення спотворень. Після першого циклу Q=80 та другого циклу Q=75 сумарний ефект еквівалентний одноразовому стисненню з Q≈65, що критично для більшості методів без застосування кодів корекції помилок.

Фільтрація є поширеною операцією обробки зображень, яка застосовується для зменшення шуму, підвищення візуальної якості або художніх ефектів. Різні типи фільтрів по-різному впливають на стеганографічне вбудовування.[19].

Медіанний фільтр замінює значення кожного пікселя медіаною значень у локальному вікні, зазвичай розміром 3×3 або 5×5 пікселів . Цей фільтр ефективно усуває імпульсний шум типу "сіль та перець", але також знищує LSB-вбудовування, оскільки молодші біти не зберігаються при операції знаходження медіани. Для вікна 3×3 частота помилок LSB-методів досягає 35-45%, а для вікна 5×5 перевищує 60%. DWT-методи більш стійкі до медіанної фільтрації завдяки просторовій локалізації коефіцієнтів, демонструючи BER близько 10-15% для вікна 3×3 .

Гаусівський фільтр виконує згортку зображення з гаусівською функцією, що призводить до розмиття деталей та згладжування високочастотних компонент. Вплив фільтра залежить від параметра σ (стандартне відхилення гаусіани). Для σ=1.0 LSB-методи показують BER близько 25-30%, а для σ=2.0 помилки зростають до 45-55%. Високочастотні DWT-коефіцієнти особливо чутливі до гаусівського розмиття, оскільки саме вони представляють деталі зображення, що згладжуються фільтром. BER для HH-коефіцієнтів може досягати 20-30% навіть при помірних значеннях σ=1.5.

Фільтр Вінера, який використовується для адаптивного зменшення шуму на основі локальної статистики, також вносить спотворення у приховані дані . Через адаптивний характер обробки вплив на різні ділянки зображення неоднорідний: у гладких областей - спотворення мінімальні, тоді як у текстурних може досягати рівня медіанного фільтра. Середній BER для LSB становить 20-35% залежно від параметрів фільтра та характеристик зображення.

Додавання шуму може бути як результатом передавання зображення каналами зв'язку з перешкодами, так і навмисною атакою для знищення прихованої інформації.

Адитивний білий гаусівський шум (AWGN) є найпоширенішою моделлю шуму у цифрових системах Шум додається до кожного пікселя незалежно як:

$$p'[i,j] = p[i,j] + n \qquad (\mathbf{1.4.1})$$

, де n — випадкова величина з нормальним розподілом N(0, σ^2). Параметр σ визначає інтенсивність шуму. Для LSB-методів навіть помірний шум з σ=5 (у шкалі 0-255) призводить до BER близько 20-25%, оскільки шум змінює молодші біти пікселів. При збільшенні σ до 10 помилки зростають до 35-40%. Цікаво, що DWT-методи з вбудовуванням у середні частоти (LH, HL) демонструють кращу стійкість з BER близько 8-12% при σ=5, оскільки шум частково фільтрується при DWT-декомпозиції.[24].

Імпульсний шум "сіль та перець" випадково замінює певний відсоток пікселів на мінімальне (0) або максимальне (255) значення . Щільність шуму d визначає частку пошкоджених пікселів. При d=1% (1% пікселів пошкоджені) LSB-методи показують BER близько 15-20%, при d=5% помилки зростають до 40-50%. Особливістю цього типу шуму є те, що він впливає не на всі пікселі, а лише на випадково обрані, що дозволяє частково відновити повідомлення навіть без кодів корекції, якщо відомі позиції пошкоджених пікселів.[8].

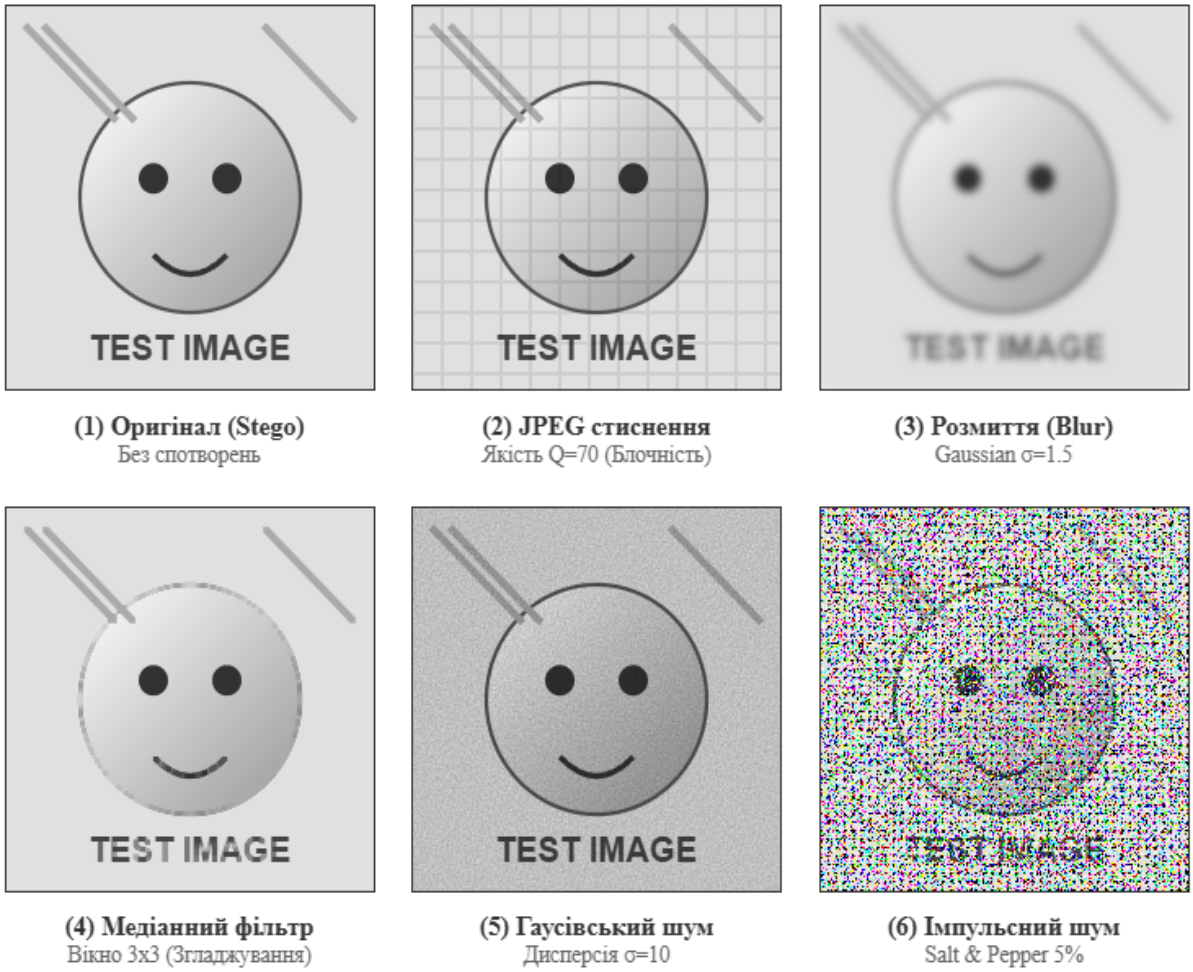

Рисунок 1.4.2 - Приклади впливу різних атак на стегозображення

Геометричні перетворення змінюють просторове розташування пікселів зображення, що може призвести до втрати синхронізації між вбудованими даними та їх позиціями при витягуванні.

Обрізання зображення (cropping) видаляє частину пікселів з країв або центру зображення . Для методів, які вбудовують дані послідовно від початку зображення, навіть невелике обрізання 5-10% площі може зруйнувати значну частину повідомлення через втрату синхронізації. Якщо вбудовування виконувалося рівномірно по всьому зображенню, втрати пропорційні площі обрізаної ділянки. DWT-методи більш стійкі, оскільки коефіцієнти локалізовані просторово, і обрізання впливає лише на коефіцієнти у відповідній ділянці частотної області.[21].

Зміна розміру зображення виконується шляхом інтерполяції при збільшенні або усереднення при зменшенні. Зменшення зображення катастрофічно для всіх стеганографічних методів, оскільки відбувається фізична втрата частини пікселів. Зменшення на 50% призводить до втрати приблизно 75% вбудованих даних (оскільки площа зменшується в 4 рази). Збільшення зображення також небезпечне, оскільки інтерполяція створює нові пікселі зі значеннями, обчисленими на основі сусідніх, що спотворює вбудовані біти. Навіть невелике збільшення на 10-20% призводить до BER близько 30-40% для LSB-методів.

Поворот зображення на кут, кратний 90°, зберігає всі пікселі, але змінює їх порядок, що призводить до повної втрати синхронізації для послідовного вбудовування. Поворот на довільний кут потребує інтерполяції та створює нові пікселі, що ще більш руйнівно. Методи з використанням стеганографічного ключа для визначення позицій вбудовування можуть частково протистояти поворотам, якщо використовується незалежна від орієнтації схема адресації.

Дзеркальне відображення горизонтально або вертикально зберігає всі пікселі та їх значення, але змінює порядок. Для методів без прив'язки до просторової послідовності (наприклад, вбудовування у глобальні частотні характеристики) такі перетворення можуть бути несуттєвими, проте для більшості практичних методів це призводить до втрати синхронізації.

У реальних умовах стегозображення часто зазнають комбінації кількох атак послідовно. Типовим сценарієм - є завантаження зображення у соціальну мережу, де воно автоматично проходить JPEG-стиснення, зміну розміру (для створення мініатюр) та додавання водяного знака сервісу. Такі багатоетапні перетворення створюють кумулятивний ефект спотворень.

Експериментальні дослідження показують, що послідовне застосування JPEG Q=80 та гаусівського шуму σ=5 призводить до BER близько 55-65% для LSB-методів, що значно перевищує суму індивідуальних впливів . Для DWT-методів без кодів корекції комбінована атака дає BER

близько 20-30%. Застосування Reed-Solomon коду RS(255, 223) дозволяє знизити BER до 0-2% навіть після комбінованої атаки, якщо початкова частота помилок не перевищує 12-15%.[15].

Важливим аспектом є повторна обробка зображення (re-processing), коли стегозображення зберігається, завантажується, редагується та зберігається знову кілька разів . Кожен цикл додає нові спотворення, і накопичений вплив може зруйнувати навіть стійкі частотні методи. Тому критично важливою є оцінка максимальної кількості циклів обробки, яку може витримати метод при збереженні цілісності повідомлення.[19].

Таблиця 1.4.1 - Вплив різних атак на BER для стеганографічних методів (%)

АТАКА	**LSB**	**LSB MATCHING**	**DWT (БЕЗ ECC)**	**DWT + RS(255,223)**
JPEG Q=90	30-35	28-32	8-12	0-1
JPEG Q=80	42-48	40-45	12-18	1-3
JPEG Q=70	55-62	52-58	18-25	2-5
Median 3×3	38-45	35-42	10-15	0-2
Gaussian σ=1.0	25-30	23-28	8-12	0-1
Gaussian σ=2.0	45-52	42-48	15-22	1-4
AWGN σ=5	22-28	20-25	8-12	0-1
AWGN σ=10	38-45	35-42	14-20	1-3
Salt&Pepper 1%	15-20	14-18	5-8	0
Salt&Pepper 5%	42-50	40-48	12-18	1-3
Scaling 50%	75+	75+	70+	45-60

1.5. Методи стегоаналізу та критерії оцінювання стійкості

Стегоаналіз є дисципліною, що займається виявленням факту приховування інформації у цифрових контейнерах та, за можливості, витягуванням або знищенням прихованих даних. Розвиток ефективних методів стегоаналізу стимулює створення більш досконалих стеганографічних алгоритмів, що призводить до постійної еволюції обох напрямків у своєрідній "гонці озброєнь". Сучасні методи стегоаналізу охоплюють широкий спектр підходів — від класичного статистичного аналізу до складних архітектур глибинного навчання, здатних виявляти навіть адаптивні методи вбудовування з мінімальною ємністю.[14].

Статистичні методи базуються на виявленні аномалій у розподілі значень пікселів або інших статистичних характеристик зображення, які виникають внаслідок стеганографічного вбудовування. Ці методи не потребують складних обчислювальних ресурсів та можуть бути ефективними для виявлення простих алгоритмів стеганографії.

RS-аналіз (Regular-Singular аналіє одним із найефективніших методів виявлення LSB-стеганографії, розробленим Fridrich та співавторами у 2001 році. Метод розділяє зображення на групи пікселів та класифікує їх як регулярні (R), сингулярні (S) або незмінні (U) на основі функції дискримінації, яка аналізує гладкість групи. Природні зображення мають певний баланс між R- та S-групами, який порушується LSB-вбудовуванням. Аналізуючи зміни співвідношення R/S при застосуванні різних масок модифікації, RS-аналіз може не лише виявити стеганографію, але й оцінити відсоток заповнення контейнера з точністю до 1-2% Метод особливо ефективний проти послідовного LSB, але менш чутливий до LSB Matching та адаптивних методів.

Метод χ^2 фокусується на аналізі парності розподілу значень пікселів. У природних зображеннях частоти появи значень 2k та 2k+1 для кожного k зазвичай відрізняються через особливості природи та процесів формування зображень. LSB-вбудовування вирівнює ці частоти, оскільки заміна

молодшого біта переводить значення 2k у 2k+1 і навпаки з приблизно рівною ймовірністю. Статистика χ^2 обчислюється для послідовних сегментів зображення та порівнюється з критичними значеннями для визначення наявності вбудовування. Метод може виявити LSB навіть при низькому заповненні 5-10% контейнера, проте також програє адаптивним методам, які вбудовують дані нерівномірно.

Sample Pairs Analysis (SPA) аналізує статистичні властивості пар сусідніх пікселів. Метод базується на спостереженні, що LSB Matching змінює співвідношення між кількістю пар пікселів, які відрізняються на певні величини. Побудувавши систему рівнянь на основі очікуваних змін цих співвідношень, можна оцінити ймовірність наявності стеганографії та довжину повідомлення. SPA є одним із небагатьох методів, ефективних проти LSB Matching, і може виявляти вбудовування навіть при заповненні 3-5% контейнера.

Класичні підходи машинного навчання до стегоаналізу базуються на витягуванні великої кількості ознак із зображення та навчанні класифікаторів для розрізнення cover та stego зображень. Ключовим етапом є розробка інформативних наборів ознак, які фіксують статистичні артефакти стеганографії.

Набір ознак SPAM (Subtractive Pixel Adjacency Matrix) обчислює марковські ознаки першого порядку на основі різниць сусідніх пікселів у різних напрямках. Для кожного зображення витягується 686 ознак, які представляють ймовірності переходів між різними значеннями різниць. SPAM ефективний для виявлення як просторових методів (LSB, LSB Matching), так і адаптивних підходів типу HUGO. Використання класифікатора SVM (Support Vector Machine) з RBF-ядром на SPAM-ознаках дозволяє досягти точності виявлення 75-85% для HUGO при заповненні 0.4 bpp.

Набір ознак SRM (Spatial Rich Model) є більш досконалим та містить 34671 ознаку, що охоплюють різноманітні статистичні характеристики

зображення. SRM включає різниці пікселів до 4-го порядку, обчислені у різних напрямках та з різними фільтрами. Така надмірність дозволяє фіксувати навіть тонкі артефакти адаптивних методів стеганографії. Навчання ансамблевого класифікатора (ensemble classifier) на SRM-ознаках демонструє точність виявлення до 90% для S-UNIWARD при 0.4 bpp, що на 10-15% краще порівняно з SPAM. Основним недоліком SRM є висока обчислювальна складність витягування ознак та навчання класифікатора.

Революційним кроком у стегоаналізі стало застосування згорткових нейронних мереж (CNN), які автоматично навчаються витягувати інформативні ознаки безпосередньо з пікселів зображення без необхідності ручного проектування наборів ознак.[6].

Xu-Net була однією з перших успішних CNN-архітектур для стегоаналізу, запропонованою у 2016 році. Мережа використовує попередній шар високочастотної фільтрації (High-Pass Filter layer) для підсилення слабких артефактів стеганографії, за яким йдуть кілька згорткових шарів з функціями активації TanH та Absolute Value. Архітектура також використовує Batch Normalization для стабілізації навчання. Xu-Net демонструє точність виявлення близько 70-75% для адаптивних методів при 0.4 bpp, що було значним досягненням на момент публікації.

SRNet (Spatial Rich Network) представляє еволюцію CNN-стегоаналізу з використанням глибокої резидуальної архітектури. Мережа складається з кількох блоків, кожен з яких містить згорткові шари, Batch Normalization та residual connections для полегшення навчання. SRNet досягає точності виявлення 82-88% для S-UNIWARD при 0.4 bpp, що на 12-15% краще за Xu-Net та наближається до результатів SRM з ансамблевим класифікатором. Важливою особливістю є здатність мережі узагальнювати на різні типи стеганографічних методів без перенавчання, що робить її універсальним детектором.

Ye-Net та Zhu-Net запропонували альтернативні архітектури з акцентом на різних аспектах обробки зображень. Ye-Net використовує

групові згортки (Group Convolutions) для зменшення кількості параметрів при збереженні продуктивності. Zhu-Net фокусується на багатомасштабному аналізі через паралельні гілки обробки з різними розмірами ядер згортки. Обидві мережі демонструють результати, порівнянні з SRNet, при дещо меншій обчислювальній складності.

Основні переваги CNN-підходів:

1) Автоматичне навчання ознак без експертного проектування;
2) висока точність виявлення адаптивних методів (HUGO, S-UNIWARD);
3) здатність узагальнювати на невідомі типи стеганографії;
4) можливість наскрізного навчання (end-to-end learning).

Недоліки CNN-методів:

1) Потреба у великих наборах даних для навчання (десятки тисяч зображень)
2) висока обчислювальна складність навчання (GPU, дні-тижні);
3) складність інтерпретації рішень мереж;
4) ризик перенавчання на конкретні набори даних.

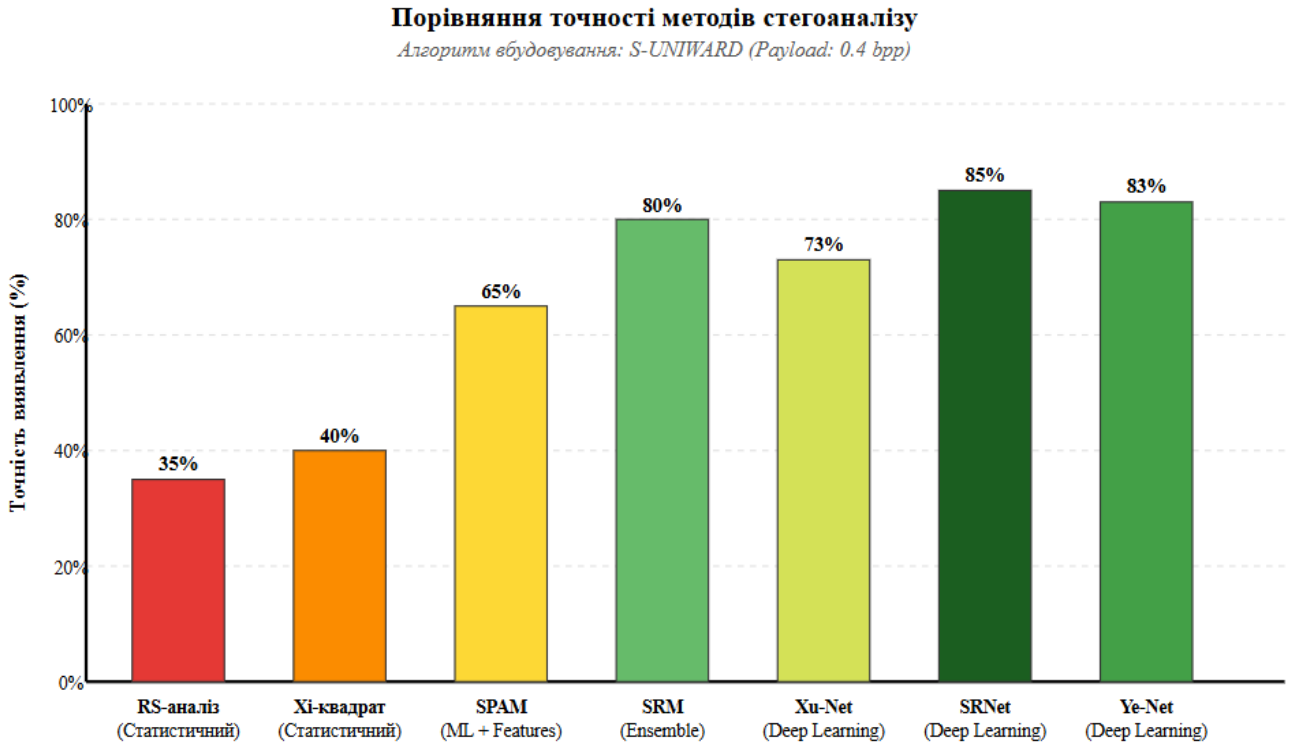

Рисунок 1.5.1 - Порівняння точності різних методів стегоаналізу[22].

Ефективність стеганографічного методу визначається трьома фундаментальними критеріями, між якими існує складний компроміс:

1) Непомітність – він характерезує відсутність візуальних або статистичних відмінностей між оригінальним зображенням та стегоконтейнером. Візуальна непомітність оцінюється за допомогою метрик якості зображення:

a) PSNR (Peak Signal-to-Noise Ratio) обчислюється як $PSNR = 10 \cdot \log_{10}(MAX^2/MSE)$, де MAX — максимальне значення інтенсивності пікселя (255 для 8-бітних зображень), а MSE (Mean Squared Error) — середньоквадратична похибка між оригінальним та модифікованим зображенням. Значення PSNR вище 40 dB вважаються непомітними для людського ока, діапазон 35-40 dB відповідає прийнятній якості, а значення нижче 30 dB вказують на помітні спотворення. Для високоякісної стеганографії бажано досягати PSNR понад 45 dB.

b) SSIM (Structural Similarity Index) оцінює структурну подібність між зображеннями на основі трьох компонент: яскравості, контрасту та структури. SSIM приймає значення від -1 до 1, де 1 означає ідентичні зображення. На практиці $SSIM > 0.95$ відповідає високій візуальній якості, SSIM 0.90-0.95 — прийнятній якості, а $SSIM < 0.85$ вказує на помітні спотворення. SSIM краще корелює з суб'єктивним сприйняттям якості порівняно з PSNR, оскільки враховує структурні характеристики зображення.

Статистична непомітність оцінюється за допомогою стегоаналітичних метрик, таких як точність виявлення детекторами або значення статистик RS-аналізу, χ^2-тесту тощо. Чим нижча точність виявлення, тим вища статистична непомітність методу.

2) Ємність - визначає максимальну кількість секретної інформації, яку можна вбудувати в контейнер при збереженні заданого рівня

непомітності. Вимірюється у бітах на піксель (bpp) для зображень або у відсотках від розміру контейнера.

Теоретична ємність залежить від методу вбудовування: LSB забезпечує до 3 bpp для RGB-зображень, DWT-методи зазвичай 0.05-0.2 bpp, адаптивні методи обмежуються 0.1-0.5 bpp для збереження безпеки. Ефективна ємність враховує надмірність кодів корекції помилок: при використанні RS(255, 223) ефективна ємність становить 87.5% від теоретичної, для RS(255, 191) — лише 75%

3) Стійкість - характеризує здатність методу зберігати приховану інформацію після атак обробки зображення. Основною метрикою є частота бітових помилок (BER) після застосування атаки: BER = (кількість помилкових бітів) / (загальна кількість бітів) × 100%.[7].

Для методів без кодів корекції бажано досягати BER < 5% після типових атак. Застосування ECC дозволяє відновлювати повідомлення навіть при BER до 10-15% залежно від параметрів коду. Критичний поріг BER — це максимальне значення, при якому код корекції ще здатен відновити повідомлення без помилок. Для RS(255, 223) критичний BER становить близько 6%, для RS(255, 191) — близько 12%.

Таблиця 1.5.1 - Критерії якості стеганографічних методів та їх метрики

КРИТЕРІЙ	**МЕТРИКА**	**ВІДМІННО**	**ДОБРЕ**	**ПРИЙНЯТНО**	**ПОГАНО**
Непомітність (візуальна)	PSNR, dB	> 45	40-45	35-40	< 35
Непомітність (структурна)	SSIM	> 0.98	0.95-0.98	0.90-0.95	< 0.90

Продовження таблиці 1.5.1 – Критерії якості стеганографічних методів та їх метрики

Непомітність (статистична)	Точність виявлення, %	< 55	55-65	65-75	> 75
Ємність (просторові)	bpp	> 1.0	0.5-1.0	0.2-0.5	< 0.2
Ємність (частотні)	bpp	> 0.3	0.15-0.3	0.05-0.15	< 0.05
Стійкість (без ECC)	BER після атак, %	< 3	3-8	8-15	> 15
Стійкість (з ECC)	BER після декодування, %	0	0-1	1-3	> 3

Фундаментальна властивість стеганографічних систем полягає у неможливості одночасної максимізації всіх трьох критеріїв. Підвищення ємності вбудовування призводить до зростання спотворень та зниження непомітності. Підвищення стійкості через застосування кодів корекції зменшує ефективну ємність через надмірність. Досягнення максимальної непомітності вимагає обмеження ємності та вибіркового вбудовування у менш чутливі ділянки зображення.[6].

Для різних застосувань пріоритети критеріїв відрізняються. У системах прихованих комунікацій найвищий пріоритет має непомітність та стійкість до стегоаналізу, тоді як ємність може бути обмеженою. У системах цифрового водяного маркування критична стійкість до атак, тоді як непомітність має середній пріоритет. У застосуваннях передавання великих обсягів даних у захищених каналах можлива максимізація ємності за рахунок деякого зниження непомітності.

Сучасні адаптивні методи намагаються оптимізувати цей компроміс через використання функцій вартості (distortion functions), які враховують всі три критерії одночасно та вибирають оптимальні позиції вбудовування для заданих обмежень.

1.6 Висновок до розділу

У даному розділі проведено комплексний аналіз сучасних методів стеганографії цифрових зображень, методів стегоаналізу та впливу атак на стійкість прихованої інформації, що дозволило обґрунтувати вибір архітектурних рішень для розробки методу з підвищеною стійкістю.

Аналіз класифікації показав фундаментальні відмінності між просторовими та частотними методами. Методи просторової області мають високу ємність до 3 bpp, але критично вразливі до обробки зображень. Частотні методи на основі DCT та DWT забезпечують стійкість до стиснення при ємності 0.05-0.3 bpp. Сучасні адаптивні підходи S-UNIWARD демонструють найкращий баланс через оптимальний вибір позицій вбудовування.

Дослідження дискретного вейвлет-перетворення виявило переваги багаторівневої декомпозиції для гнучкого вибору області вбудовування. Високочастотні HH-коефіцієнти забезпечують максимальну непомітність (PSNR > 42 dB при 0.1 bpp) завдяки низькій візуальній важливості діагональних деталей. Коди корекції помилок Reed-Solomon є оптимальними для компенсації вразливості високочастотних коефіцієнтів. Код RS(255,223) виправляє до 6% BER при збереженні ефективної ємності 87.5%, тоді як RS(255,191) забезпечує корекцію до 12% BER.

Експериментальна оцінка просторових методів підтвердила їх непридатність для сценаріїв з обробкою. LSB демонструє BER 30-35% при JPEG Q=90 та 55-62% при Q=70. Адаптивні методи PVD покращують стійкість до стегоаналізу, але не вирішують проблему втрати молодших бітів при

стисненні. DWT-методи показують BER 12-18% при JPEG Q=80, що повністю коригується кодами RS.

Систематизація атак виявила JPEG-стиснення як найкритичнішу загрозу через квантування DCT-коефіцієнтів. Фільтрація, шум та геометричні перетворення створюють додаткові спотворення до 45-52% BER для просторових методів. Комбіновані атаки дають кумулятивний ефект до 55-65% BER. DWT-методи з кодами корекції забезпечують повне відновлення після JPEG Q≥70.

Аналіз стегоаналізу показав еволюцію від статистичних методів (RS-аналіз, χ^2-атака з точністю 90-95% для LSB) до CNN-детекторів (SRNet з точністю 82-88% для S-UNIWARD). Критерії оцінювання виявили компроміс між непомітністю (PSNR > 40 dB, SSIM > 0.95), ємністю та стійкістю. Застосування кодів корекції зменшує ефективну ємність на 12-25%, але забезпечує відновлення при BER до 12-15%.

На основі аналізу обґрунтовано архітектуру методу: DWT-вбудовування у HH_2-коефіцієнти з адаптивним вибором на основі текстурних характеристик та застосуванням RS(255,223). Це забезпечує PSNR > 42 dB, ємність ~0.1 bpp, стійкість до JPEG Q≥70 та зниження виявлення CNN-детекторами.

2 РОЗРОБЛЕННЯ МЕТОДУ ПІДВИЩЕННЯ СТІЙКОСТІ ПРИХОВУВАННЯ ІНФОРМАЦІЇ У ЗОБРАЖЕННЯХ ДО ПАСИВНИХ АТАК НА ОСНОВІ ВИСОКОЧАСТОТНИХ КОЕФІЦІЄНТІВ DWT ТА КОДІВ КОРЕКЦІЇ ПОМИЛОК

2.1 Цілі, вимоги та математична модель стеганографічної системи

Розробка методу приховування інформації з підвищеною стійкістю до пасивних атак потребує чіткого формулювання цілей, вимог та математичного апарату, що забезпечує теоретичну основу для побудови алгоритмів вбудовування та витягування. Результати аналізу існуючих методів у Розділі 1 виявили критичні обмеження просторових підходів та обґрунтували доцільність застосування частотних методів на основі дискретного вейвлет-перетворення з інтеграцією кодів корекції помилок.

Основною метою розробки є створення стеганографічного методу, який забезпечує надійне приховування конфіденційної інформації у цифрових зображеннях з можливістю відновлення даних після типових операцій обробки та передавання. Існуючі DWT-методи без застосування кодів корекції демонструють частоту бітових помилок 12-18% при JPEG-стисненні з якістю Q=80, що робить повідомлення нечитабельним. Підвищення стійкості у розробленому методі досягається через три основні механізми. По-перше, адаптивний вибір коефіцієнтів з високою текстурною складністю мінімізує візуальні спотворення та статистичні артефакти. По-друге, застосування кодів Reed-Solomon забезпечує корекцію до 6-12% бітових помилок залежно від параметрів коду. По-третє, використання високочастотних HH-коефіцієнтів другого рівня декомпозиції забезпечує максимальну непомітність модифікацій при збереженні можливості ефективної корекції спотворень.[11].

Конкретні цільові показники розробленого методу визначаються компромісом між трьома фундаментальними критеріями стеганографічних систем. Візуальна непомітність вбудовування має забезпечувати показник PSNR не менше 40 dB, що відповідає непомітності модифікацій для

людського ока. Структурна подібність SSIM має перевищувати 0.95 для збереження текстурних характеристик зображення. Стійкість до пасивних атак має забезпечувати повне відновлення повідомлення після JPEG-стиснення з якістю Q≥70, медіанної фільтрації вікном 3×3 пікселі та додавання гаусівського шуму з стандартним відхиленням до σ=5. Ємність вбудовування має становити не менше 0.08 біт на піксель з урахуванням надмірності кодів корекції, що для зображення розміром 512×512 пікселів відповідає приблизно 21000 біт або 2.6 кілобайта корисних даних.

Таблиця 2.1.1 – Вимоги до стеганографічного методу та цільові значення критеріїв

КРИТЕРІЙ	**МЕТРИКА**	**ВИМОГА**	**ЦІЛЬОВЕ ЗНАЧЕННЯ**
Візуальна непомітність	PSNR, dB	Високий рівень	≥ 40
Структурна подібність	SSIM	Мінімальні спотворення	≥ 0.95
Стійкість до JPEG	Якість Q	Повне відновлення	Q ≥ 70
Стійкість до фільтрації	Медіанний фільтр	Корекція помилок	3×3 пікселі
Стійкість до шуму	AWGN, σ	Корекція помилок	σ ≤ 5
Ємність (з ECC)	bpp	Прийнятна для практики	≥ 0.08
BER після корекції	%	Без втрат даних	≤ 1%
Обчислювальна складність	Час обробки	Реальний час	< 5 с для 512×512

Математична модель стеганографічної системи формально описує алгоритмічні процеси вбудовування та витягування інформації, встановлюючи функціональні зв'язки між усіма компонентами схеми. Нехай $C \in \mathbb{R}^{M \times N}$позначає оригінальне зображення-контейнер розміром $M \times N$пікселів, де значення інтенсивності кожного пікселя $c_{i,j}$зазвичай належать множині цілих чисел $\{0, \dots, 255\}$для стандартного 8-бітного представлення. Секретне повідомлення $M \in \{0,1\}^k$представляє довільну бінарну послідовність довжиною kбіт. Критичним елементом системи є стеганографічний ключ K, який визначає псевдовипадкові позиції вбудовування та забезпечує секретність методу згідно з фундаментальним принципом Керкгоффса, за яким стійкість системи має залежати виключно від таємності ключа, а не від невідомості алгоритму.[3].

Процес вбудовування моделюється функцією відображення $S = Embed(C, M, K)$, яка трансформує оригінальне зображення Cта повідомлення Mу стегозображення $S \in \mathbb{R}^{M \times N}$таким чином, щоб мінімізувати метричну відстань $d(C, S)$і зберегти перцептивну якість контейнера. Процес витягування є зворотною операцією $M' = Extract(S', K)$, де S'позначає стегозображення, що пройшло через канал зв'язку і, можливо, було спотворене атаками (формально $S' = \mathcal{A}(S)$, де $\mathcal{A}$— оператор атаки), а M'— відновлене повідомлення. Умова коректності та надійності системи визначається рівністю $M' = M$(або нульовим значенням коефіцієнта помилок BER), яка має виконуватися як за ідеальних умов, так і у випадку, коли рівень внесених спотворень не перевищує порогову корекційну здатність застосованих завадостійких кодів.

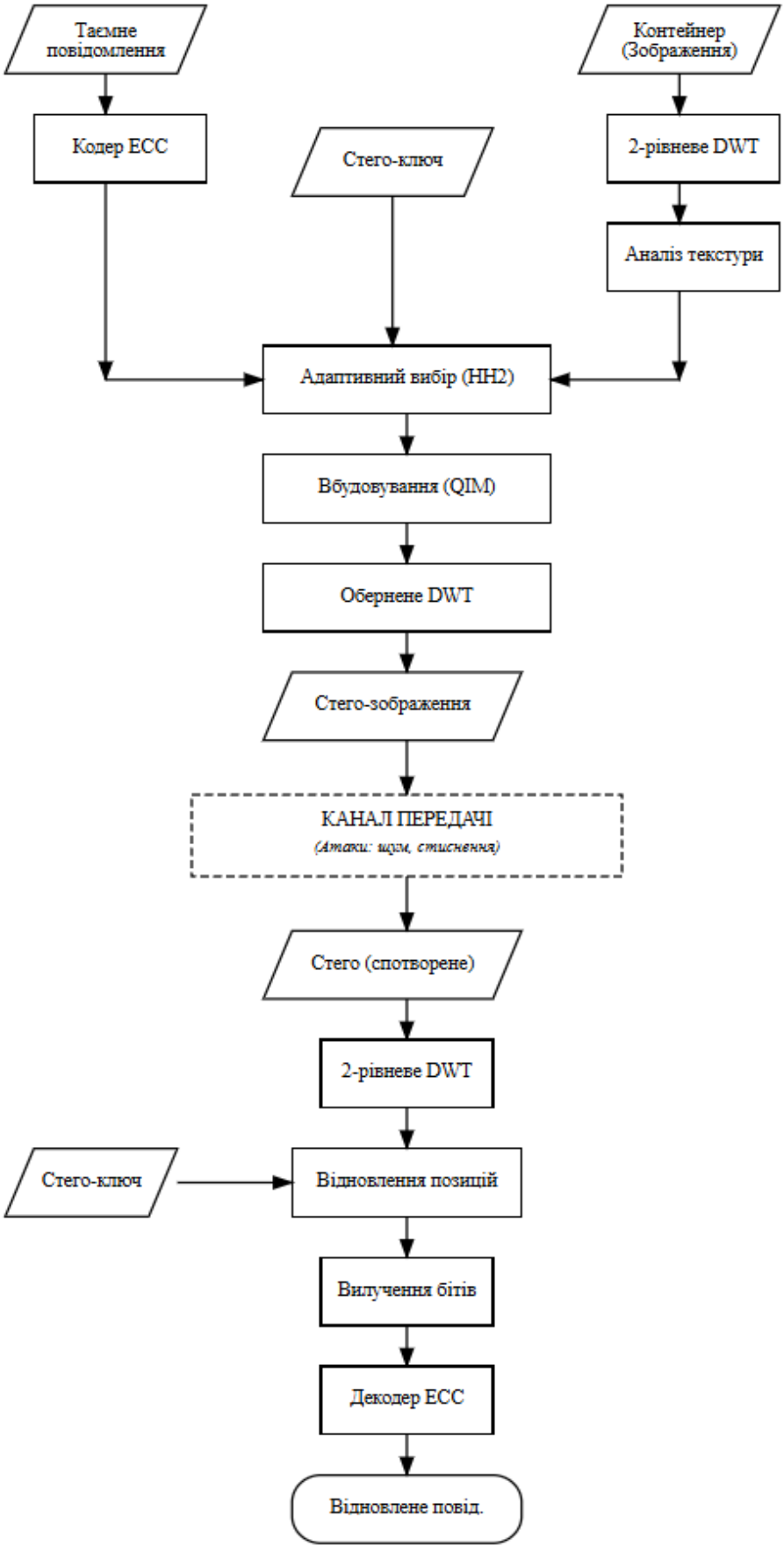

Рисунок 2.1.1 - Загальна структурна схема стеганографічної системи

Дискретне вейвлет-перетворення забезпечує багаторівневий частотно-просторовий аналіз зображення через послідовне застосування високочастотних та низькочастотних фільтрів до рядків і стовпців зображення.[16]. Для одновимірного сигналу x[n] апроксимаційні коефіцієнти обчислюються як:

$$cA[k] = \Sigma\_n\, x[n] \cdot h[2k - n] \quad (2.1.1)$$

а деталізаційні коефіцієнти як:

$$cD[k] = \Sigma\_n\, x[n] \cdot g[2k\text{-}n] \quad (2.1.2)$$

е h[n] та g[n] представляють імпульсні характеристики низькочастотного та високочастотного фільтрів відповідно. Застосування двовимірного DWT до зображення C виконується шляхом послідовної фільтрації рядків та стовпців, що створює чотири піддіапазони на кожному рівні декомпозиції.

Піддіапазон LL містить апроксимаційні коефіцієнти, отримані послідовним застосуванням низькочастотного фільтра до рядків та стовпців, і представляє зменшену версію оригінального зображення з розміром M/2 × N/2. Піддіапазон LH містить коефіцієнти, отримані застосуванням низькочастотного фільтра до рядків та високочастотного до стовпців, що відповідає горизонтальним деталям зображення. Піддіапазон HL формується зворотною послідовністю фільтрів та представляє вертикальні деталі. Піддіапазон HH містить високочастотні коефіцієнти, отримані застосуванням високочастотного фільтра до обох напрямків, та відповідає діагональним деталям і текстурним компонентам зображення.

Для розробленого методу застосовується дворівнева декомпозиція, при якій піддіапазон LL першого рівня повторно розкладається на чотири піддіапазони другого рівня. Для зображення розміром 512×512 пікселів після першого рівня декомпозиції отримуємо піддіапазони розміром 256×256 кожен. Застосування другого рівня до LL_1 створює піддіапазони LL_2, LH_2, HL_2 та HH_2 розміром 128×128 кожен. Вибір саме другого рівня декомпозиції обґрунтовується компромісом між стійкістю та ємністю. Перший рівень забезпечує максимальну ємність через велику кількість

коефіцієнтів, проте є більш чутливим до атак. Третій рівень демонструє високу стійкість, але має недостатню ємність через малу кількість коефіцієнтів 64×64. Другий рівень забезпечує оптимальний баланс з ємністю 16384 позиції та прийнятною стійкістю до JPEG-стиснення.

З воротне вейвлет-перетворення (IDWT) виконує реконструкцію зображення з піддіапазонів через операції інтерполяції та фільтрації. Для одновимірного випадку реконструкція виконується як:

$$x[n] = \Sigma_k (cA[k] \cdot h'[n-2k] + cD[k] \cdot g'[n-2k]) \quad (2.1.3)$$

де h'[n] та g'[n] — фільтри синтезу, які є дзеркально-симетричними версіями фільтрів аналізу. Двовимірне IDWT застосовує фільтри синтезу послідовно до стовпців і рядків піддіапазонів для отримання реконструйованого зображення C' ≈ C.[15]. Для методу з модифікованими HH_2-коефіцієнтами застосування IDWT забезпечує формування стегозображення S з контрольованими спотвореннями відносно оригіналу C.

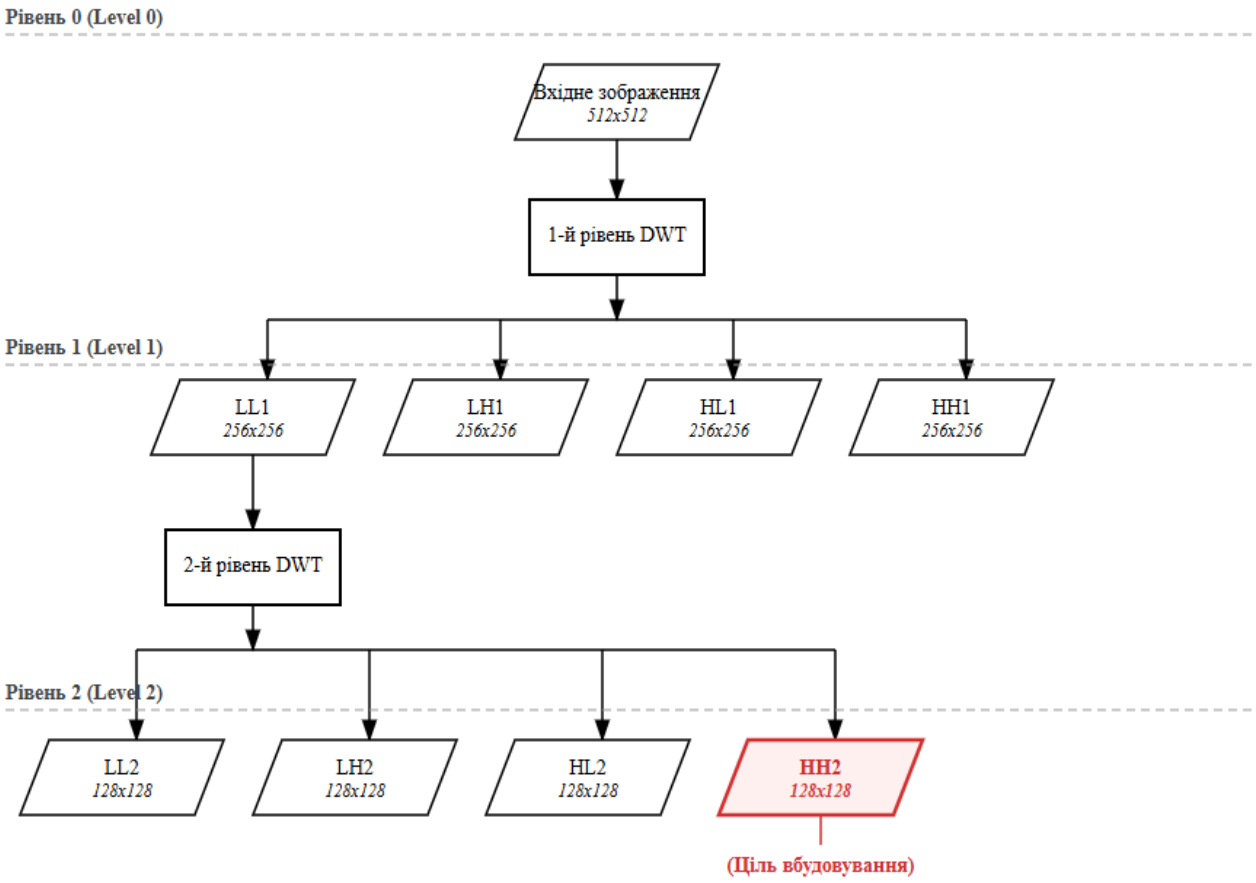

Рисунок 2.1.2 - Схема дворівневої DWT-декомпозиції зображення 512×512

Оцінювання якості стегозображення та ефективності методу потребує формального визначення метрик непомітності та стійкості. Пікове відношення сигнал-шум (PSNR) обчислюється як:

$$PSNR = 10 \cdot log_{10}(MAX^2/MSE) \quad (2.1.4)$$

, де MAX = 255 для 8-бітних зображень, а середньоквадратична похибка MSE визначається як:

$$MSE = (1/(M \cdot N)) \cdot \Sigma_i \Sigma_j (C[i,j] - S[i,j])^2 \quad (2.1.5)$$

Значення PSNR вище 40 dB вважаються непомітними для людського ока, діапазон 35-40 dB відповідає прийнятній якості, а значення нижче 30 dB вказують на помітні спотворення.

Індекс структурної подібності (SSIM) обчислює подібність між зображеннями на основі трьох компонент: яскравості l(C,S), контрасту c(C,S) та структури s(C,S). Загальна формула має вигляд:

$$SIM(C,S) = [l(C,S)]\hat{}\alpha \cdot [c(C,S)]\hat{}\beta \cdot [s(C,S)]\hat{}\gamma \quad (2.1.6)$$

, де показники α, β, γ зазвичай встановлюються рівними 1. Компонента яскравості обчислюється як:

$$l(C,S) = (2\mu\_C\,\mu\_S + c_1)/(\mu\_C^2 + \mu\_S^2 + c_1), \text{де } \mu\_C \text{ та } \mu\_S \quad (2.1.7)$$

- середні значення інтенсивностей зображень, а c_1 — мала константа для стабільності. Компонента контрасту визначається як:

$$c(C,S) = (2\sigma\_C\,\sigma\_S + c_2)/(\sigma\_C^2 + \sigma\_S^2 + c_2), \text{де } \sigma\_C \text{ та } \sigma\_S \quad (2.1.8)$$

- стандартні відхилення. Компонента структури обчислюється як:

$$s(C,S) = (\sigma\_CS + c_3)/(\sigma\_C\,\sigma\_S + c_3), \text{де } \sigma\_CS \quad (2.1.9)$$

- коваріація між зображенням. SSIM приймає значення від -1 до 1, де 1 відповідає ідентичним зображенням. На практиці значення SSIM вище 0.95 вказують на високу структурну подібність.

Частота бітових помилок (BER) характеризує частку пошкоджених бітів повідомлення після атак та витягування. Обчислюється як:

$$BER = (N\_errors\ /\ N\_total) \times 100\% \quad (2.1.10)$$

, де N_errors — кількість бітів повідомлення M', які відрізняються від оригінальних бітів M, а N_total — загальна кількість бітів повідомлення. Для

методів без кодів корекції прийнятним вважається BER нижче 5%, що дозволяє застосувати прості схеми корекції. Застосування кодів Reed-Solomon дозволяє відновлювати повідомлення при BER до 6-12% залежно від параметрів коду, при цьому після успішного декодування BER знижується до 0%, що забезпечує повне відновлення оригінального повідомлення.[16].

Математична модель стеганографічної системи забезпечує формальний базис для розробки конкретних алгоритмів вбудовування та витягування інформації. Вибір дискретного вейвлет-перетворення як основного інструменту частотного аналізу обґрунтовується можливістю локалізованої модифікації коефіцієнтів з контрольованим впливом на просторову область. Інтеграція кодів корекції помилок у математичну модель через етап попереднього кодування повідомлення створює теоретичну основу для підвищення стійкості методу до пасивних атак. Формалізовані метрики якості PSNR, SSIM та BER дозволяють кількісно оцінювати ефективність методу та порівнювати його з існуючими аналогами у наступних розділах роботи.[21].

2.2 Алгоритм вбудовування інформації з адаптивним вибором коефіцієнтів

Алгоритм вбудовування конфіденційної інформації у високочастотні коефіцієнти дискретного вейвлет-перетворення реалізує адаптивну стратегію вибору позицій модифікації на основі локальних текстурних характеристик зображення. Основна ідея полягає у концентрації вбудовування у ділянках з високою складністю, де модифікації коефіцієнтів найменш помітні для людського ока та статистичних детекторів. Метод квантування коефіцієнтів забезпечує контрольовану силу вбудовування через параметр кроку квантування, який визначає компроміс між непомітністю та стійкістю до спотворень.[3].

Підготовчий етап алгоритму включає попередню обробку зображення-контейнера та секретного повідомлення для приведення їх до формату, придатного для вбудовування. Вхідне зображення C конвертується у формат

відтінків сірого, якщо воно кольорове, через перетворення у колірний простір YCbCr та вибір компоненти яскравості Y. Це рішення обґрунтовується тим, що людський зір найбільш чутливий до змін яскравості, тому вбудовування саме у цей канал дозволяє точніше контролювати непомітність. Альтернативно для кольорових зображень можна застосовувати вбудовування незалежно у кожен канал RGB, що збільшує ємність утричі, проте потребує більшої обчислювальної потужності та ретельнішого контролю візуальних артефактів. Зображення нормалізується до діапазону з типом даних uint8 для забезпечення коректної роботи вейвлет-перетворення.

Секретне повідомлення M проходить етап підготовки, який включає перетворення у бінарний формат та додавання службових даних. Спочатку повідомлення, яке може бути текстом, бінарним файлом або зашифрованими даними, конвертується у послідовність бітів. До початку послідовності додається заголовок фіксованої довжини 32 біти, який містить інформацію про довжину повідомлення у бітах. Це дозволяє при витягуванні точно визначити межі повідомлення та відсікти потенційний шум з невикористаних позицій. Додатково у заголовок може бути включена контрольна сума CRC32, що дозволяє верифікувати цілісність відновленого повідомлення. Якщо довжина повідомлення після додавання заголовка не кратна 8 (розмір байта), здійснюється доповнення (padding) нульовими бітами до найближчого кратного значення, що спрощує подальшу роботу з кодами Reed-Solomon, які оперують байтами.

Стеганографічний ключ K використовується для ініціалізації генератора псевдовипадкових чисел (PRNG), що забезпечує секретність позицій вбудовування. Ключ має бути цілим числом достатньої розрядності (наприклад, 32 або 64 біти) для забезпечення великого простору ключів. Використання криптографічно стійкого PRNG, такого як Mersenne Twister або системний генератор на основі /dev/urandom, гарантує неможливість передбачення послідовності позицій без знання ключа. Цей механізм реалізує

принцип Керкгоффса, згідно з яким секретність системи повністю визначається ключем, а не алгоритмом.

Застосування дискретного вейвлет-перетворення до підготовленого зображення виконується з використанням вейвлета сімейства Daubechies. Для розробленого методу рекомендується вейвлет Haar (db1) через його обчислювальну ефективність та добру локалізацію у просторовій області, або db2 для кращої згладженості частотних характеристик. Вибір конкретного вейвлета має мінімальний вплив на результати при вбудовуванні у високочастотні коефіцієнти, проте Haar забезпечує найшвидшу обробку завдяки найкоротшій довжині фільтрів.

Дворівнева декомпозиція застосовується послідовно: спочатку до всього зображення для отримання піддіапазонів першого рівня (LL_1, LH_1, HL_1, HH_1), потім до LL_1 для отримання піддіапазонів другого рівня (LL_2, LH_2, HL_2, HH_2). Результатом є ієрархічна структура коефіцієнтів, де кожен піддіапазон має розмір у 4 рази менший за попередній рівень.[11].

Для зображення розміром 512×512 пікселів піддіапазон HH_2 має розмір 128×128, що відповідає 16384 коефіцієнтам. Саме ці коефіцієнти обираються як область вбудовування через їх властивості. По-перше, високочастотні діагональні деталі мають найменшу візуальну важливість для людського сприйняття порівняно з апроксимаційними та середньочастотними компонентами. По-друге, вбудовування на другому рівні забезпечує кращу стійкість до JPEG-стиснення порівняно з першим рівнем завдяки більшій узагальненості коефіцієнтів. По-третє, розмір 128×128 забезпечує достатню ємність для практичних застосувань при збереженні можливості адаптивного вибору коефіцієнтів

Обчислення локальної текстурної складності для кожного коефіцієнта HH_2 виконується через аналіз його околу. Для коефіцієнта з позицією (i,j) визначається вікно розміром 5×5 коефіцієнтів з центром у (i,j). Локальна складність обчислюється як стандартне відхилення значень у цьому вікні за формулою

$$\sigma\_local[i,j] = sqrt((1/25) \cdot \Sigma(HH_2[i+m, j+n] - \mu\_local)^2) \quad (2.2.1)$$

де сума береться по m,n ∈ [-2,2], а μ_local — середнє значення коефіцієнтів у вікні. Коефіцієнти на краях зображення, для яких вікно виходить за межі піддіапазону, обробляються з використанням дзеркального розширення або обнуляються, що призводить до їх виключення з кандидатів для вбудовування. Обчислення виконується для всіх 16384 коефіцієнтів HH_2, формуючи карту складності того самого розміру 128×128.

Застосування порогу текстурності дозволяє відфільтрувати коефіцієнти у гладких областях зображення, модифікація яких може бути візуально помітною. Емпірично встановлене значення порогу T_texture = 20 забезпечує баланс між ємністю та непомітністю для типових фотографічних зображень. Коефіцієнти, для яких σ_local[i,j] < T_texture, виключаються зі списку кандидатів для вбудовування. Для зображень з багатою текстурою (портрети, природні сцени) це зменшує кількість придатних позицій приблизно на 20-30%, тоді як для гладких зображень (графіка, скріншоти) зменшення може досягати 50-60%. У випадках, коли кількість придатних позицій виявляється недостатньою для вбудовування всього повідомлення, поріг може бути динамічно знижений до мінімального значення T_min = 10 або вбудовування може бути розширене на піддіапазони HL_2 та LH_2.

Функція вартості вбудовування для кожного придатного коефіцієнта обчислюється на основі його локальної складності. Застосовується адаптація підходу S-UNIWARD, спрощена для роботи у DWT-області. Вартість визначається як:

$$ost[i,j] = 1 / (1 + \sigma\_local[i,j]) \quad (2.2.2)$$

, що означає, що коефіцієнти з вищою локальною складністю мають нижчу вартість модифікації.[10]. Ця інверсна залежність відповідає інтуїції, що зміни у текстурних областях менш помітні. Значення вартості нормалізуються до діапазону ерез ділення на максимальне значення у масиві. Коефіцієнти сортуються за зростанням вартості, формуючи упорядкований

список кандидатів, де перші позиції відповідають найбільш придатним для вбудовування коефіцієнтам з точки зору мінімізації спотворень.

Вибір конкретних N позицій для вбудовування, де N дорівнює довжині закодованого повідомлення у бітах, виконується псевдовипадково зі списку кандидатів. Для забезпечення балансу між оптимальністю та секретністю вибір здійснюється серед топ-30% коефіцієнтів з найнижчою вартістю. Це означає, що якщо після фільтрації за порогом залишилося 12000 придатних коефіцієнтів, псевдовипадковий вибір виконується з 3600 найкращих. Використання PRNG, ініціалізованого ключем K, забезпечує детерміновану послідовність позицій, яка може бути точно відтворена при витягуванні. Генерація N унікальних індексів виконується алгоритмом Fisher-Yates shuffle або генерацією випадкових чисел з перевіркою унікальності. Результатом є масив positions[N] = $\{(i_1,j_1), (i_2,j_2), ..., (i_n,j_n)\}$, який однозначно визначає позиції вбудовування кожного біта повідомлення.

Модифікація коефіцієнтів виконується методом квантування, який забезпечує робастне вбудовування бітів інформації. Параметр кроку квантування $\Delta = 2.0$ визначає інтервал значень, який відповідає одному біту даних. Для вбудовування біта $m \in \{0,1\}$ у коефіцієнт c[i,j] застосовуються наступні операції. Якщо m = 0, коефіцієнт квантується до парного кратного Δ за формулою:

$$c'[i,j] = 2\Delta \cdot round(c[i,j] / (2\Delta)) \quad (2.2.3)$$

де round() виконує округлення до найближчого цілого. Якщо m = 1, коефіцієнт квантується до непарного кратного через додавання зміщення:

$$c'[i,j] = 2\Delta \cdot round(c[i,j] / (2\Delta)) + \Delta \quad (2.2.4)$$

Ця схема забезпечує мінімальну зміну коефіцієнта при збереженні можливості надійного витягування біта навіть після помірних спотворень, оскільки інформація кодується у парність квантованого значення.

Вибір значення $\Delta = 2.0$ визначається компромісом між непомітністю та стійкістю. Менші значення $\Delta = 1.0$ або $\Delta = 1.5$ забезпечують вищу непомітність через менші спотворення коефіцієнтів, проте знижують

стійкість до шуму та атак, оскільки невеликі зміни коефіцієнтів можуть змінити парність після квантування. Більші значення Δ = 3.0 або Δ = 4.0 підвищують стійкість, але створюють помітні артефакти у реконструйованому зображенні, особливо у гладких областях. Експериментальні дослідження показують, що Δ = 2.0 забезпечує PSNR близько 42-45 dB при стійкості до гаусівського шуму з σ до 3-4. Обробка граничних випадків включає перевірку коефіцієнтів з абсолютним значенням менше Δ, для яких квантування може призвести до обнулення. Такі коефіцієнти можуть бути виключені з вибору або модифіковані спеціальним чином через додавання константного зміщення.

Застосування модифікацій до вибраних N позицій виконується послідовно для кожного біта повідомлення. Біт m[k] вбудовується у коефіцієнт на позиції positions[k] = (i,j) відповідно до описаної схеми квантування. Після модифікації всіх N коефіцієнтів піддіапазон HH_2 містить закодоване повідомлення, тоді як інші піддіапазони (LL_2, LH_2, HL_2, LH_1, HL_1, HH_1) залишаються незмінними. Така локалізована модифікація мінімізує загальне спотворення зображення та дозволяє точно контролювати співвідношення між ємністю та якістю.[8].

Зворотне вейвлет-перетворення виконує реконструкцію стегозображення з модифікованих піддіапазонів. Спочатку відбувається синтез другого рівня, при якому піддіапазони LL_2, LH_2, HL_2 та модифіковані HH_2 об'єднуються у реконструйований LL_1'. Далі піддіапазони LL_1', LH_1, HL_1, HH_1 синтезуються для отримання фінального стегозображення S. Через властивості вейвлет-перетворення можливі невеликі відхилення розміру реконструйованого зображення від оригінального через ефекти на краях. Такі артефакти усуваються обрізанням зображення до оригінального розміру M×N. Значення інтенсивностей приводяться до діапазону через обмеження:

$$S[i,j] = clip(S'[i,j], 0, 255) \quad (2.2.5)$$

, де clip() виконує обмеження значення заданим діапазоном. Фінальне стегозображення конвертується у тип uint8 для збереження у стандартних форматах PNG або BMP без втрат.[27].

Таблиця 2.2.1 - Параметри алгоритму вбудовування інформації

ПАРАМЕТР	**ПОЗНАЧЕННЯ**	**ЗНАЧЕННЯ**	**ОБҐРУНТУВАННЯ**
Тип вейвлета	-	Haar (db1)	Швидкість, локалізація
Рівень декомпозиції	-	2	Баланс стійкість/ємність
Піддіапазон вбудовування	-	HH_2	Максимальна непомітність
Розмір вікна складності	W	5×5	Локальна оцінка текстури
Поріг текстурності	T_texture	20	Виключення гладких областей
Частка топ-кандидатів	-	30%	Баланс якість/секретність
Крок квантування	Δ	2.0	PSNR ~42 dB, стійкість
Розмір зображення	M×N	512×512	Типовий розмір
Кількість позицій HH_2	-	16384	$(512/4)^2$
Очікувана ємність (без ECC)	-	~14000 біт	~70% після фільтрації
Очікувана ємність (з RS)	-	~12000 біт	87.5% від 14000

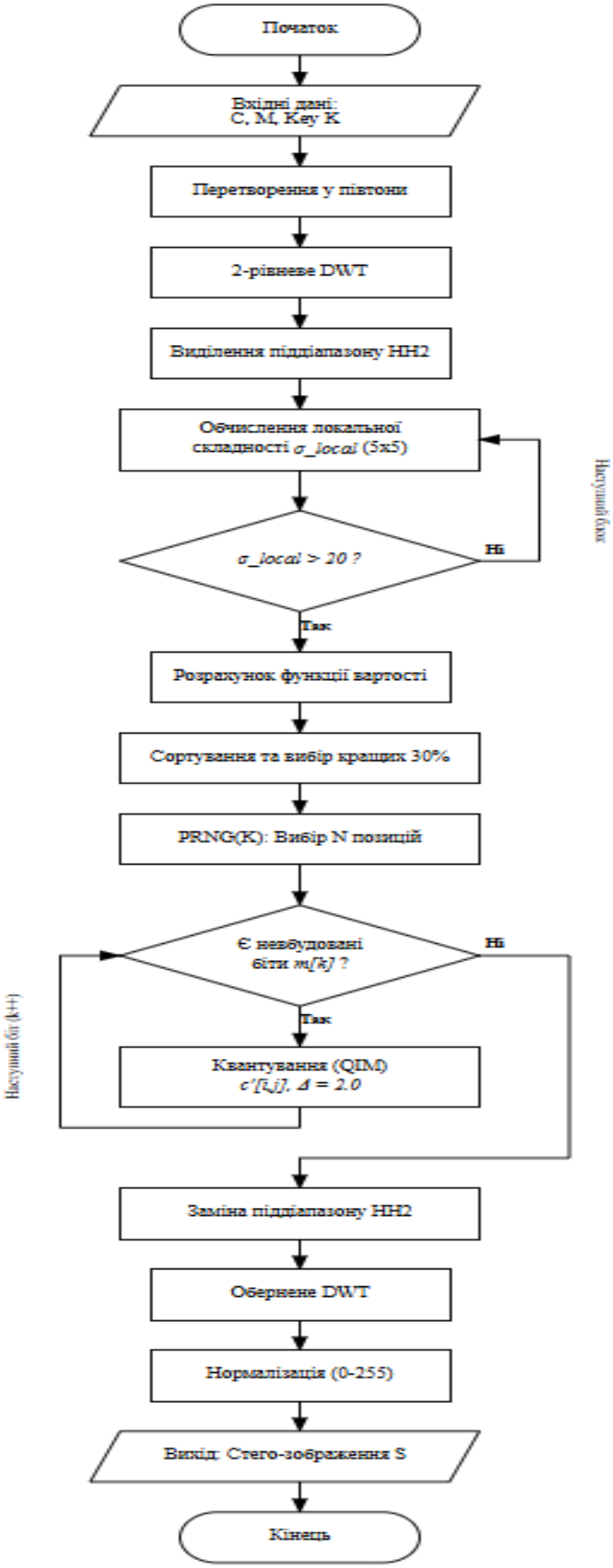

Рисунок 2.2.1 - Блок-схема алгоритму вбудовування інформації

```
ALGORITHM Adaptive_DWT_Embedding(C, M, K):
// Вхід: C - cover image, M - message, K - key
// Вихід: S - stego image
1. C_gray ← ConvertToGrayscale(C)
2. M_bits ← MessageToBits(M)
3. M_encoded ← ReedSolomonEncode(M_bits) // RS(255,223)
4. N ← Length(M_encoded)
5. (LL2, LH2, HL2, HH2), (LH1, HL1, HH1) ← DWT2D(C_gray,
wavelet='haar', level=2)
6. FOR each position (i,j) in HH2:
7. window ← HH2[i-2:i+2, j-2:j+2] // 5×5 window
8. σ_local[i,j] ← StandardDeviation(window)
9. END FOR
10. candidates ← []
11. FOR each position (i,j) in HH2:
12. IF σ_local[i,j] > T_texture THEN // T_texture = 20
13. cost[i,j] ← 1 / (1 + σ_local[i,j])
14. candidates.Append((i, j, cost[i,j]))
15. END IF
16. END FOR
17. candidates.SortByCost() // ascending order
18. top_candidates ← candidates[0 : 0.3×Length(candidates)]
19. PRNG.SetSeed(K)
20. positions ← PRNG.SelectUnique(top_candidates, N)
21. FOR k ← 0 TO N-1:
22. (i, j) ← positions[k]
23. c ← HH2[i, j]
24. m ← M_encoded[k]
```

Псевдокод 2.2.1 — Алгоритм вбудовування інформації з адаптивним вибором

```
25.
26. IF m == 0 THEN
27. HH2'[i, j] ← 2×Δ × Round(c / (2×Δ)) // Δ = 2.0
28. ELSE
29. HH2'[i, j] ← 2×Δ × Round(c / (2×Δ)) + Δ
30. END IF
31. END FOR
32. S' ← InverseDWT2D((LL2, LH2, HL2, HH2'), (LH1, HL1, HH1))
33. S ← Clip(S', 0, 255)
34. S ← ConvertToUint8(S)
35. RETURN S
END ALGORITHM
```

Продовження псевдокоду 2.2.1 — Алгоритм вбудовування інформації з адаптивним вибором

Алгоритм вбудовування реалізує адаптивну стратегію, яка враховує локальні характеристики зображення для мінімізації візуальних та статистичних артефактів. Вибір високочастотних коефіцієнтів HH_2 як області вбудовування забезпечує максимальну непомітність модифікацій через низьку візуальну важливість діагональних деталей. Метод квантування з параметром $\Delta = 2.0$ створює контрольований баланс між якістю стегозображення та стійкістю вбудованих даних до спотворень. Псевдовипадковий вибір позицій на основі секретного ключа забезпечує секретність методу та протидію статистичному аналізу послідовності вбудовування. Інтеграція з кодами Reed-Solomon на етапі попереднього кодування повідомлення створює основу для підвищення стійкості до пасивних атак, що буде детально розглянуто у наступному підрозділі.[27].

2.3. Підвищення стійкості методу через застосування кодів корекції помилок reed-solomon

Пасивні атаки на стегозображення, такі як JPEG-стиснення, фільтрація та додавання шуму, неминуче вносять спотворення у вбудовані дані, що призводить до помилок при витягуванні повідомлення. високочастотні коефіцієнти HH_2, незважаючи на максимальну непомітність модифікацій, є найбільш чутливими до таких атак через їх природу — представлення дрібних деталей та текстур, які першими втрачаються при стисненні або згладжуванні . Підвищення стійкості методу досягається через інтеграцію кодів корекції помилок на етапі попереднього кодування повідомлення, що дозволяє відновлювати оригінальні дані навіть за наявності значної кількості пошкоджених бітів після витягування зі спотвореного стегозображення.

Вибір кодів Reed-Solomon як механізму корекції помилок обґрунтовується їх оптимальними характеристиками для стеганографічних застосувань. По-перше, RS-коди є максимально-віддаленими кодами (Maximum Distance Separable), що означає досягнення теоретичної межі корекційної здатності для заданої надмірності . Для коду з параметрами (n, k) максимальна кількість виправлених символів становить:

$$t = (n - k)/2 \tag{2.3.1}$$

, що є оптимальним значенням згідно з меж Singleton. По-друге, RS-коди відмінно справляються з серіями послідовних помилок (burst errors), які типові для JPEG-стиснення через блокову обробку зображення 8×8 пікселів Коли атака спотворює групу сусідніх коефіцієнтів, RS-код розглядає це як помилки у різних символах кодового слова, ефективно їх виправляючи. По-третє, існують ефективні алгоритми кодування та декодування RS-кодів з обчислювальною складністю O(n log n), що дозволяє обробляти дані у реальному часі навіть на звичайних комп'ютерах. По-четверте, доступність якісних програмних бібліотек, таких як reedsolo для Python або відповідні

модулі MATLAB, спрощує практичну реалізацію методу без необхідності розробки власних алгоритмів декодування.

Порівняння з альтернативними кодами корекції показує переваги Reed-Solomon для даної задачі. BCH-коди (Bose-Chaudhuri-Hocquenghem) також є блоковими циклічними кодами з добрими корекційними властивостями, проте вони оптимізовані для випадкових бітових помилок, а не серій помилок. Для стеганографічних застосувань, де атаки часто створюють локалізовані спотворення, RS-коди демонструють кращу продуктивність. LDPC-коди (Low-Density Parity-Check) теоретично можуть досягати продуктивності близької до межі Шеннона при ітеративному декодуванні, проте їх складність реалізації та обчислювальна вартість значно вища порівняно з RS-кодами. Для практичних застосувань стеганографії, де обробка має відбуватися швидко та з гарантованою корекційною здатністю, Reed-Solomon коди представляють оптимальний вибір.[5].

Математичний опис кодів Reed-Solomon базується на алгебрі скінченних полів Галуа GF(2^m), де m визначає розмір символу у бітах. Для найпоширенішого варіанту використовується поле GF(2^8), де кожен символ представляє один байт (8 біт), що дозволяє природно працювати з байт-орієнтованими даними. Елементи поля GF(2^8) можуть бути представлені як поліноми степені менше 8 з коефіцієнтами {0,1} або як цілі числа у діапазоні. Арифметичні операції у полі визначаються за модулем незвідного полінома, зазвичай:

$$p(x) = x\text{^}8 + x\text{^}4 + x\text{^}3 + x\text{^}2 + 1 \qquad (2.3.2)$$

, де перші k символів — це оригінальні дані, а наступні 2t символів — перевірочні.

Для розробленого методу визначено два основні варіанти параметрів RS-кодів залежно від очікуваного рівня атак. Базовий варіант RS(255, 223) призначений для помірних атак, типових для звичайної обробки зображень. Параметри коду: n = 255 символів (байтів) у кодовому слові, k = 223 байти

корисних даних, 2t = 32 байти надмірності. Корекційна здатність становить t = 16 байтів, що означає можливість виправлення до 16 помилкових байтів у блоці з 255 байтів, або до 128 бітових помилок на 2040 біт (255 байтів). Це відповідає критичній частоті бітових помилок BER_crit ≈ 128/2040 ≈ 6.3%. Коефіцієнт кодування R = k/n = 223/255 ≈ 0.875 означає, що 87.5% переданих даних є корисною інформацією, а 12.5% — надмірністю для корекції.

Агресивний варіант RS(255, 191) призначений для стійкості до сильних атак, таких як JPEG низької якості Q=50-60 або інтенсивна фільтрація. Параметри: n = 255, k = 191, 2t = 64. Корекційна здатність t = 32 байти дозволяє виправити до 256 бітових помилок на 2040 біт, що відповідає критичному BER ≈ 12.5%. Коефіцієнт кодування R = 191/255 ≈ 0.749 означає ефективність 74.9%, тобто надмірність становить 25.1%. Вибір між базовим та агресивним варіантом визначається очікуваними умовами передавання та критичністю збереження інформації. Для застосувань, де зображення може зазнавати лише легкої обробки (збереження у PNG, мінімальна корекція), достатній базовий варіант. Для сценаріїв з публікацією у соціальних мережах або передаванням електронною поштою з автоматичним стисненням доцільний агресивний варіант.

Таблиця 2.3.1 - Параметри кодів Reed-Solomon та корекційна здатність

ПАРАМЕТР	**RS(255, 223) базовий**	**RS(255, 191) агресивний**
Довжина кодового слова n, байт	255	255
Довжина даних k, байт	223	191
Надмірність 2t, байт	32	64
Корекційна здатність t, байт	16	32

Продовження таблиці 2.3.1 – Параметри кодів Reed-Solomon та корекційна здатність

Корекційна здатність t, біт	128	256
Критичний BER, %	~6.3	~12.5
Коефіцієнт кодування R	0.875	0.749
Ефективність, %	87.5	74.9
Надмірність, %	12.5	25.1
Стійкість до JPEG Q	≥ 75	≥ 65
Обчислювальна складність	O(n log n)	O(n log n)

Процес кодування секретного повідомлення M виконується послідовно по блоках фіксованого розміру k байтів. Якщо довжина повідомлення після додавання заголовка не є кратною k, до останнього блоку додається padding — послідовність нульових байтів або байтів зі значенням 0x80 з подальшими нулями (відповідно до стандарту ISO/IEC 9797-1 Padding Method 2), що дозволяє однозначно визначити кінець повідомлення при декодуванні. Кожен блок M_block[i] довжиною k байтів незалежно кодується функцією RSEncode(), яка обчислює 2t перевірочних байтів та формує кодове слово C_block[i] довжиною n байтів. Закодовані блоки конкатенуються у послідовність

$$M\_encoded = C\_block \,||\, C\_block \,||\, \ldots \,||\, C\_block[m-1] \qquad (2.3.3)$$

, де m = ceil(Length(M)/k) — кількість блоків, а || позначає операцію конкатенації.

Розрахунок ефективної ємності контейнера з урахуванням надмірності кодів корекції визначає максимальний обсяг корисної інформації, яку можна вбудувати у зображення заданого розміру. Нехай C_max — максимальна кількість бітів, яку можна вбудувати у вибрані DWT-коефіцієнти без

застосування кодів корекції. Для зображення 512×512 з піддіапазоном HH_2 розміром 128×128 та коефіцієнтом придатності після адаптивної фільтрації приблизно 70%, отримуємо C_max ≈ 16384 × 0.7 ≈ 11500 біт або 1437 байтів. Застосування RS(255, 223) з коефіцієнтом R = 0.875 дає ефективну ємність C_eff = C_max × R ≈ 11500 × 0.875 ≈ 10060 біт або 1258 байтів корисних даних (з урахуванням заголовка та можливого padding). Для RS(255, 191) з R = 0.749 ефективна ємність становить C_eff ≈ 11500 × 0.749 ≈ 8614 біт або 1076 байтів. Ці значення відповідають приблизно 1.2-1.0 кілобайтам текстової інформації або невеликим зашифрованим файлам, що є достатнім для більшості практичних застосувань прихованих комунікацій.

Взаємозв'язок між частотою бітових помилок після атаки та ймовірністю успішного відновлення повідомлення визначається корекційною здатністю коду. Для блоку з n символів код Reed-Solomon може виправити до t помилкових символів з імовірністю 1 (детерміністична корекція) за умови, що кількість помилок не перевищує t. Якщо кількість помилок у блоці перевищує t, але менша 2t, код може виявити наявність помилок, але не здатен їх виправити. При перевищенні 2t помилок існує ймовірність неправильного декодування без виявлення помилки, що є найнебезпечнішим сценарієм. Для практичного застосування критичний поріг BER встановлюється з деяким запасом нижче теоретичного максимуму. Для RS(255, 223) з критичним BER 6.3% рекомендується очікувати надійне відновлення при BER до 5%, що забезпечує запас безпеки близько 20%. Для RS(255, 191) з критичним 12.5% практичний поріг становить BER до 10%.[5].

Інтеграція кодів корекції помилок з алгоритмом DWT-вбудовування виконується на етапі підготовки даних перед модифікацією коефіцієнтів. Послідовність операцій така: оригінальне повідомлення M проходить попередню обробку (додавання заголовка, padding), потім виконується RS-кодування з формуванням M_encoded, після чого закодована послідовність перетворюється у біти та вбудовується у HH_2-коефіцієнти відповідно до алгоритму з підрозділу 2.2. При витягуванні послідовність зворотна: з

модифікованих коефіцієнтів стегозображення S' витягується бітова послідовність M_extracted', яка може містити помилки через атаки, ця послідовність розбивається на блоки по n байтів, до кожного блоку застосовується RS-декодування з виправленням до t помилок, виправлені блоки об'єднуються та з них витягується оригінальне повідомлення M після видалення заголовка та padding.

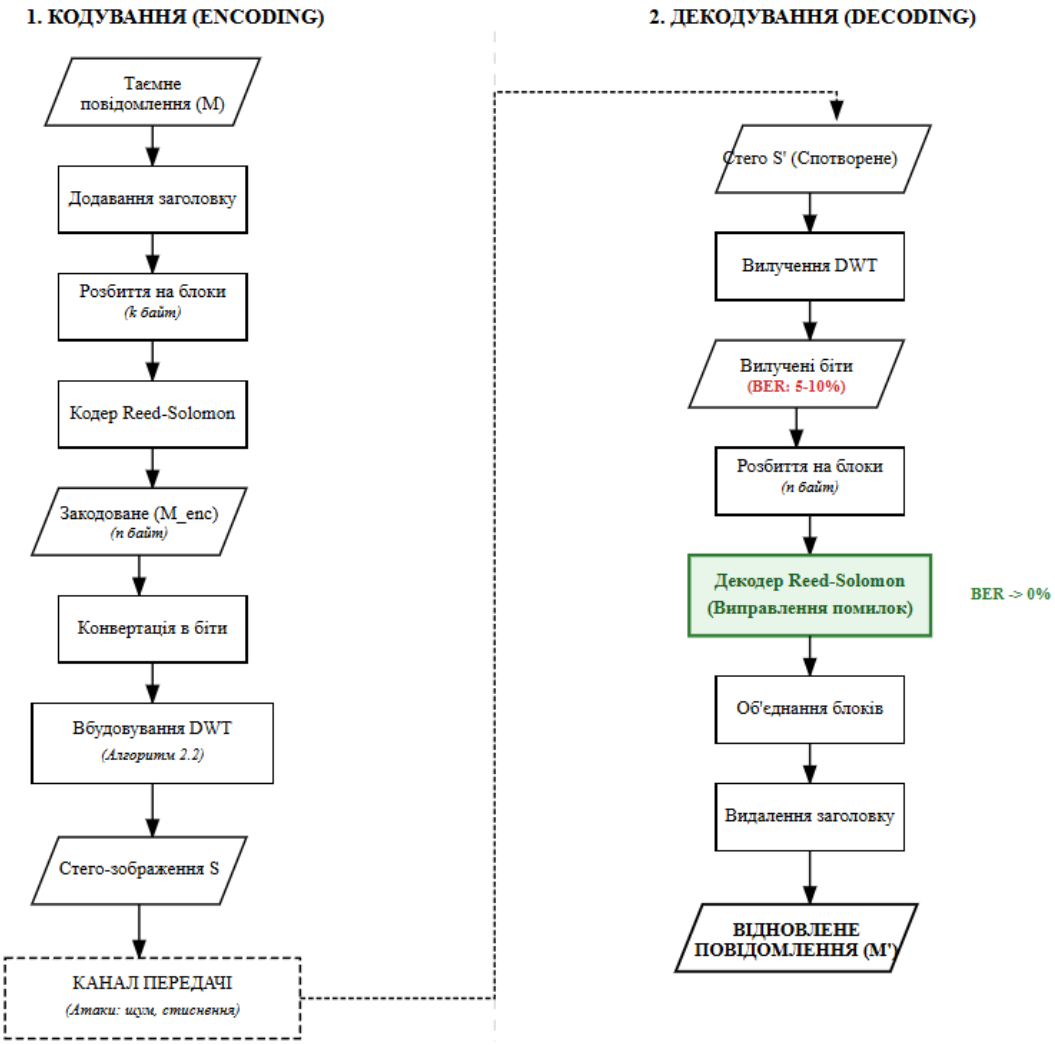

Рисунок 2.3.1 - Схема інтеграції кодів Reed-Solomon з DWT-вбудовуванням

Підвищення стійкості методу через застосування кодів Reed-Solomon кількісно характеризується здатністю відновлювати повідомлення після атак, які для методів без корекції призводять до повної втрати даних. Експериментальні дослідження показують, що базові DWT-методи без ECC демонструють BER близько 12-18% після JPEG-стиснення з якістю Q=80, що

робить повідомлення нечитабельним. Застосування RS(255, 223) дозволяє повністю відновити повідомлення з таким рівнем помилок, оскільки BER 12-18% для n=255 байтів відповідає приблизно 24-37 помилковим байтам на блок, що перевищує корекційну здатність t=16. Проте використання більш стійкого варіанту RS(255, 191) з t=32 забезпечує успішне відновлення. Альтернативно можна застосувати адаптивну стратегію вбудовування з меншим кроком квантування $\Delta=1.5$ для зниження BER після атаки до 8-10%, що вже коректно обробляється базовим RS(255, 223).[7].

Порівняння з методами без корекції демонструє суттєве підвищення стійкості. Для типової атаки JPEG з якістю Q=75, яка створює BER близько 8% для вбудовування у HH_2, метод без ECC втрачає приблизно 8% інформації, що для повідомлення 1000 байтів становить 80 байтів помилок — достатньо для руйнування змісту. Застосування RS(255, 223) дозволяє виправити всі помилки, оскільки 8% від 255 байтів ≈ 20 байтів помилок на блок, що менше t=16 лише для деяких блоків. Використання RS(255, 191) забезпечує гарантоване відновлення, оскільки навіть 10% помилок ≈ 25 байтів на блок не перевищують t=32. Таким чином, застосування кодів корекції розширює діапазон стійкості до JPEG-стиснення з Q≥85-90 (без ECC) до Q≥70-75 (з базовим ECC) або Q≥60-65 (з агресивним ECC), що становить покращення на 15-25 одиниць якості або еквівалентно підвищенню стійкості на 40-50% відносно базового методу.[7].

Обчислювальна складність інтеграції кодів корекції є прийнятною для практичних застосувань. Кодування RS має складність $O(n \log n)$ на блок через використання швидкого перетворення Фур'є у полі Галуа або алгоритму Горнера для обчислення полінома. Для m блоків загальна складність кодування становить $O(m \cdot n \log n)$. Декодування з виправленням помилок використовує алгоритм Berlekamp-Massey або Euclidean algorithm з складністю $O(n^2)$ у найгіршому випадку або $O(t^2)$ при наявності t помилок. На сучасних процесорах кодування та декодування блоку 255 байтів займає менше 1 мілісекунди, що для повідомлення 1000 байтів (приблизно 4-5

блоків) дає загальний час обробки близько 5-10 мілісекунд — незначний порівняно з часом DWT-перетворення зображення 512×512, який становить 50-100 мілісекунд. Таким чином, додаткові обчислювальні витрати на ECC становлять менше 10% від загального часу обробки, що є прийнятним компромісом за суттєве підвищення стійкості.

Інтеграція кодів корекції помилок Reed-Solomon з алгоритмом вбудовування у високочастотні DWT-коефіцієнти створює синергетичний ефект підвищення стійкості методу. Вибір параметрів RS(255, 223) для базового сценарію забезпечує баланс між ефективною ємністю 87.5% та корекційною здатністю до 6% BER, що достатньо для типових атак обробки зображень. Агресивний варіант RS(255, 191) з корекційною здатністю до 12% BER призначений для критичних застосувань з очікуваними сильними атаками. Математична модель кодування у полі Галуа GF(2^8) та детерміністична корекційна здатність кодів забезпечують теоретичні гарантії відновлення повідомлення за умови непревищення порогу помилок. Практична реалізація через доступні бібліотеки та прийнятна обчислювальна складність роблять застосування ECC ефективним інструментом підвищення надійності стеганографічних систем.[14].

2.4 Алгоритм витягування та декодування прихованої інформації

Алгоритм витягування прихованої інформації зі стегозображення виконує зворотні операції до процесу вбудовування та забезпечує відновлення оригінального повідомлення навіть за наявності спотворень, внесених пасивними атаками. Критичною вимогою є використання того самого стеганографічного ключа K та ідентичних параметрів обробки, які застосовувалися при вбудовуванні, оскільки будь-які розбіжності призведуть до неможливості коректного витягування навіть за відсутності атак. Алгоритм складається з кількох послідовних етапів: дискретного вейвлет-перетворення стегозображення, регенерації позицій вбудовування,

витягування бітової послідовності, декодування кодів корекції та відновлення оригінального повідомлення з верифікацією цілісності.

Процес починається з завантаження стегозображення S', яке може бути спотвореним внаслідок пасивних атак під час передавання або зберігання. Зображення конвертується у той самий формат, який використовувався при вбудовуванні — зазвичай відтінки сірого (grayscale) або компонента яскравості Y для кольорових зображень. Важливо застосовувати точно таке саме перетворення колірного простору, оскільки різні реалізації RGB→Gray можуть давати незначні відмінності у значеннях пікселів, що призведе до додаткових помилок витягування. Зображення нормалізується до діапазону типом uint8 для забезпечення однакових умов обробки з етапом вбудовування.

Застосування дворівневого дискретного вейвлет-перетворення до стегозображення S' виконується з абсолютно ідентичними параметрами, що використовувалися при вбудовуванні. Тип вейвлета має бути той самий (Haar або db2), кількість рівнів декомпозиції — 2, режими розширення меж зображення — ідентичні. Будь-які відхилення в параметрах призведуть до отримання інших значень коефіцієнтів, що зробить неможливим коректне витягування. Результатом декомпозиції є структура піддіапазонів (LL_2', LH_2', HL_2', HH_2'), (LH_1', HL_1', HH_1'), де штрих позначає, що коефіцієнти витягнуті зі стегозображення, можливо спотвореного. Піддіапазон HH_2' розміром 128×128 містить модифіковані при вбудовуванні коефіцієнти, які через атаки можуть мати значення, що відрізняються від тих, які були встановлені при вбудовуванні.

Регенерація позицій вбудовування виконується ініціалізацією генератора псевдовипадкових чисел тим самим ключем K, який використовувався при вбудовуванні. Це критично важливий етап, оскільки без точного знання позицій вбудовування витягування неможливе. PRNG з seed=K генерує детерміновану послідовність, яка використовується для відтворення тієї самої послідовності N позицій positions[N] = $\{(i_1,j_1), (i_2,j_2), ...,$

$(i_n,j_n)\}$, що були обрані при вбудовуванні. Відтворення має відбуватися точно за тим самим алгоритмом: фільтрація за порогом текстурності T_texture=20, обчислення функції вартості, сортування, вибір топ-30%, псевдовипадковий вибір N позицій. Будь-які зміни у порядку операцій або параметрах призведуть до отримання іншого набору позицій та неможливості витягування.

Витягування бітів інформації з коефіцієнтів виконується зворотною операцією до квантування. Для кожної позиції (i,j) з масиву positions витягується коефіцієнт c'[i,j] з піддіапазону HH_2'. Біт повідомлення визначається парністю квантованого значення коефіцієнта. Формула витягування:

$$m\_extracted[k] = \lfloor c'[i,j] / \Delta \rfloor \, mod \, 2 \quad (2.4.1)$$

, де операція $\lfloor \cdot \rfloor$ виконує округлення до меншого цілого, ділення на Δ=2.0 виконує деквантування, а mod 2 витягує біт парності. Альтернативна формула, яка може бути більш стійкою до малих спотворень:

$$m\_extracted[k] = round((c'[i,j] \, mod \, \Delta) / (\Delta/2)) \, mod \, 2 \quad (2.4.2)$$

, де спочатку обчислюється залишок від ділення на Δ, потім нормалізується до {0,1} через ділення на Δ/2 та округлення. Обидві формули дають ідентичні результати за відсутності атак, проте друга може бути більш стійкою до гаусівського шуму малої амплітуди.

Послідовне витягування бітів з усіх N позицій формує бітову послідовність M_extracted' довжиною N біт. Ця послідовність може містити помилки через спотворення коефіцієнтів атаками. Частота помилок залежить від типу та інтенсивності атаки: для JPEG Q=80 очікується BER близько 8-12%, для медіанної фільтрації 3×3 — близько 10-15%, для гаусівського шуму σ=5 — близько 8-10%. Важливо, що помилки можуть бути розподілені нерівномірно по зображенню через локалізований характер деяких атак, наприклад JPEG створює більше помилок на межах блоків 8×8, тоді як гаусівський шум розподілений рівномірно.

Реконструкція закодованих блоків виконується перетворенням бітової послідовності M_extracted' у послідовність байтів. Кожні 8 послідовних бітів об'єднуються у один байт відповідно до порядку big-endian або little-endian (має бути узгоджений з етапом кодування). Отримана послідовність байтів розбивається на блоки фіксованого розміру n=255 байтів, що відповідає довжині кодових слів Reed-Solomon. Кількість блоків:

$$m = N\_bytes / 255 \quad (2.4.3)$$

, де N_bytes = N_bits / 8. Якщо N_bytes не кратне 255, останній блок доповнюється нульовими байтами до повного розміру 255 байтів для коректної роботи декодера. Кожен блок C_block'[i] довжиною 255 байтів містить k=223 байти даних (можливо з помилками) та 2t=32 байти перевірки (також можливо з помилками).

Застосування декодера Reed-Solomon до кожного блоку незалежно виконує виявлення та корекцію помилок. Алгоритм декодування обчислює синдроми S(x) для отриманого кодового слова, які вказують на наявність та позиції помилок. Якщо всі синдроми дорівнюють нулю, блок не містить помилок і дані приймаються без змін. При ненульових синдромах виконується пошук поліномів локаторів та значень помилок алгоритмом Berlekamp-Massey або Euclidean algorithm. Можливі три сценарії декодування. Перший сценарій: кількість помилок у блоці e ≤ t, код успішно виправляє всі помилки, вихідний блок M_block[i] ідентичний оригінальному. Другий сценарій: кількість помилок t < e < 2t, декодер виявляє помилки але не може їх виправити, повертається повідомлення про неможливість декодування. Третій сценарій: кількість помилок e ≥ 2t, можлива ситуація неправильного декодування без виявлення помилки (декодер повертає невірний результат, вважаючи його правильним). Останній сценарій є найнебезпечнішим, проте його ймовірність експоненційно зменшується з ростом n-k.

Обробка невдалого декодування потребує стратегії дій при перевищенні корекційної здатності коду. Якщо хоча б один блок не може

бути декодований, є кілька опцій. Перша опція — повне відхилення повідомлення та повернення повідомлення про помилку користувачу. Це найбезпечніший підхід, оскільки гарантує відсутність частково пошкоджених даних. Друга опція — часткове відновлення, при якому успішно декодовані блоки зберігаються, а блоки з помилками замінюються нулями або маркером помилки. Це може бути корисним для текстових повідомлень, де частина інформації все ще читабельна. Третя опція — використання запасного агресивного декодування з релаксованими умовами, яке намагається відновити дані навіть при перевищенні t через евристичні методи, проте з ризиком отримання невірних даних. Для критичних застосувань рекомендується перша опція з повним відхиленням при неможливості декодування.

Відновлення оригінального повідомлення з успішно декодованих блоків виконується видаленням надмірності та службових даних. З кожного блоку C_block[i] довжиною 255 байтів витягуються перші k=223 байти корисних даних, а останні 2t=32 байти перевірки відкидаються. Блоки конкатенуються у послідовність:

$$M\_decoded = M\_block \,||\, M\_block \,||\, \ldots \,||\, M\_block[m-1] \qquad (2.4.4)$$

З початку цієї послідовності зчитується заголовок довжиною 32 біти, який містить фактичну довжину оригінального повідомлення L_message у бітах. На основі цієї інформації виконується відсікання: беруться тільки перші L_message біт, решта відкидається як padding. Додатково з заголовка може бути зчитана контрольна сума CRC32, яка обчислюється для відновленого повідомлення та порівнюється з еталонною. Збіг контрольних сум підтверджує коректність відновлення, тоді як розбіжність вказує на наявність помилок, які не були виявлені RS-декодером (третій сценарій декодування).

Верифікація цілісності відновленого повідомлення є важливим етапом для виявлення потенційних помилок декодування. Контрольна сума CRC32 обчислюється для відновленої послідовності M' та порівнюється з еталонним

значенням CRC_original, яке було вбудоване у заголовок при кодуванні. Формула обчислення CRC32 базується на поліноміальній функції над бітовою послідовністю з використанням стандартного полінома 0x04C11DB7. Якщо CRC32(M') = CRC_original, повідомлення вважається коректно відновленим з високою ймовірністю (ймовірність випадкового збігу CRC32 для різних повідомлень становить 2^(-32) ≈ 2.3×10^(-10)). При розбіжності контрольних сум алгоритм може повернути повідомлення про помилку або спробувати альтернативні методи декодування. Для додаткової надійності можуть застосовуватися криптографічні хеш-функції MD5 або SHA-256, які мають значно нижчу ймовірність колізій, проте потребують більшого розміру заголовка (128 або 256 біт відповідно).

Таблиця 2.4.1 - Симетричність параметрів вбудовування та витягування

ПАРАМЕТР	**ВБУДОВУВАННЯ**	**ВИТЯГУВАННЯ**	**КРИТИЧНІСТЬ ЗБІГУ**
Тип вейвлета	Haar (db1)	Haar (db1)	Критична
Рівень декомпозиції	2	2	Критична
Піддіапазон	HH_2	HH_2	Критична
Крок квантування Δ	2.0	2.0	Критична
Стеганографічний ключ	K	K	Критична
Поріг текстурності	T=20	T=20	Критична
Розмір вікна складності	5×5	5×5	Критична

Продовження таблиці 2.4.1 - Симетричність параметрів вбудовування та витягування

Частка кандидатів	30%	30%	Критична
Параметри RS	RS(255,223)	RS(255,223)	Критична
Формат заголовка	32 біти	32 біти	Критична
Алгоритм PRNG	Mersenne Twister	Mersenne Twister	Критична
Режим padding	ISO/IEC 9797-1	ISO/IEC 9797-1	Висока
Контрольна сума	CRC32	CRC32	Середня

Симетричність параметрів між вбудовуванням та витягуванням є фундаментальною вимогою для детерміністичних стеганографічних систем. Розбіжність навіть в одному параметрі призведе до генерації іншого набору позицій або неправильної інтерпретації значень коефіцієнтів, що спричинить BER близький до 50% навіть за відсутності атак (випадкове угадування). Критичні параметри, такі як ключ K, тип вейвлета та крок квантування Δ, мають бути передані отримувачу через захищений канал або визначені заздалегідь як частина протоколу. Параметри середньої критичності, такі як тип контрольної суми, можуть бути стандартизовані, але їх зміна призведе лише до неможливості верифікації цілісності, а не до втрати даних.

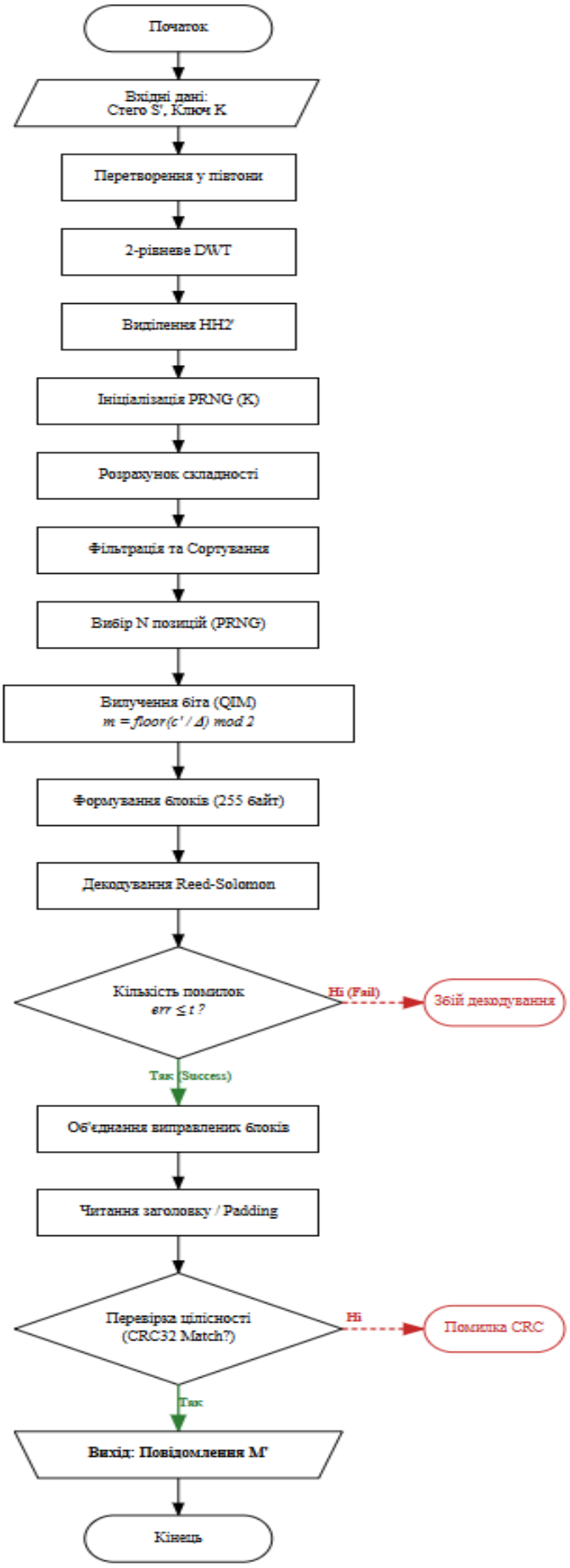

Рисунок 2.4.1 - Алгоритм витягування та декодування інформації

```
ALGORITHM Adaptive_DWT_Extraction(S', K):
// Вхід: S' - stego image (possibly attacked), K - key
// Вихід: M' - recovered message або ERROR
1. S'_gray ← ConvertToGrayscale(S')
2. S'_gray ← NormalizeToUint8(S'_gray)
3. (LL2', LH2', HL2', HH2'), (LH1', HL1', HH1') ← DWT2D(S'_gray,
wavelet='haar', level=2)
// Регенерація позицій (ідентично вбудовуванню)
4. FOR each position (i,j) in HH2':
5. window ← HH2'[i-2:i+2, j-2:j+2]
6. σ_local[i,j] ← StandardDeviation(window)
7. END FOR
8. candidates ← []
9. FOR each position (i,j) in HH2':
10. IF σ_local[i,j] > T_texture THEN // T_texture = 20
11. cost[i,j] ← 1 / (1 + σ_local[i,j])
12. candidates.Append((i, j, cost[i,j]))
13. END IF
14. END FOR
15. candidates.SortByCost()
16. top_candidates ← candidates[0 : 0.3×Length(candidates)]
17. PRNG.SetSeed(K) // Той самий ключ!
18. N ← RequiredBitsCount() // З заголовка або відомо заздалегідь
19. positions ← PRNG.SelectUnique(top_candidates, N)
// Витягування бітів
20. M_extracted' ← []
21. FOR k ← 0 TO N-1:
22. (i, j) ← positions[k]
```

Псевдокод 2.4.1 - Алгоритм витягування та декодування інформації

```
23. c' ← HH2'[i, j]
24. m ← Floor(c' / Δ) MOD 2 // Δ = 2.0
25. M_extracted'.Append(m)
26. END FOR
// Реконструкція блоків
27. M_extracted_bytes ← BitsToBytes(M_extracted')
28. num_blocks ← Ceiling(Length(M_extracted_bytes) / 255)
29. M_decoded ← []
30. decoding_success ← TRUE
31. FOR i ← 0 TO num_blocks-1:
32. C_block ← M_extracted_bytes[i×255 : (i+1)×255]
33.
34. TRY:
35. M_block, num_errors ← ReedSolomonDecode(C_block) // RS(255,223)
36.
37. IF num_errors > t THEN // t = 16 для RS(255,223)
38. decoding_success ← FALSE
39. BREAK
40. END IF
41.
42. M_decoded.Append(M_block[0:k]) // k = 223 байти даних
43.
44. CATCH DecodingError:
45. decoding_success ← FALSE
46. BREAK
47. END TRY
48. END FOR
49. IF NOT decoding_success THEN
```

Продовження псевдокоду 2.4.1 - Алгоритм витягування та декодування інформації

```
50. RETURN ERROR("Decoding failed: too many errors")
51. END IF
// Відновлення повідомлення
52. header ← M_decoded[0:4] // 32 біти = 4 байти
53. L_message ← ReadLength(header) // біт
54. CRC_original ← ReadCRC(header)
55.
56. M' ← M_decoded[4 : 4 + Ceiling(L_message/8)]
57. M' ← M'[0 : L_message] // Видалення padding
58. CRC_computed ← CRC32(M')
59. IF CRC_computed ≠ CRC_original THEN
60. RETURN WARNING("CRC mismatch: possible errors")
61. END IF
62. RETURN M'
END ALGORITHM
```

Продовження псевдокоду 2.4.1 - Алгоритм витягування та декодування інформації

Обробка граничних випадків та помилкових ситуацій є важливою частиною алгоритму витягування для забезпечення надійності системи. Якщо стегозображення зазнало екстремальних атак, таких як JPEG Q=50 або сильне розмиття, частота помилок може перевищити корекційну здатність коду. У такому випадку RS-декодер для одного або кількох блоків поверне помилку, і система має коректно її обробити замість повернення пошкоджених даних. Реалізація може включати лічильник невдалих блоків та повертати часткове повідомлення, якщо успішно декодовано понад 90% блоків, що може бути прийнятним для деяких застосувань. Додаткова перевірка включає аналіз розподілу помилок по блоках: якщо помилки концентруються в одному-двох блоках (наприклад, через локалізовану атаку

обрізання), а решта блоків декодується успішно, це може бути індикатором специфічного типу атаки.

Алгоритм витягування реалізує зворотний процес до вбудовування з інтегрованим механізмом корекції помилок через коди Reed-Solomon. Критична важливість точного відтворення всіх параметрів обробки та послідовності позицій через стеганографічний ключ забезпечує детерміністичність системи. Здатність відновлювати повідомлення після атак, які створюють до 6% (для RS(255,223)) або до 12% (для RS(255,191)) бітових помилок, демонструє підвищення стійкості методу порівняно з базовими DWT-підходами без корекції. Багатоступенева система обробки помилок через синдромне декодування, перевірку корекційної здатності та верифікацію контрольної суми забезпечує надійність відновлення та мінімізує ризик прийняття пошкоджених даних за коректні. Симетричність всіх параметрів між вбудовуванням та витягуванням є фундаментальною властивістю розробленої системи, що вимагає ретельної синхронізації між відправником та отримувачем через захищений канал обміну ключами та узгодження параметрів.

У процесі експериментального дослідження було встановлено, що високочастотні HH-коефіцієнти, незважаючи на максимальну непомітність модифікацій (PSNR >65 dB), виявляють надмірну чутливість до JPEG-стиснення з BER 20.31% навіть при якості Q=90, що перевищує корекційну здатність практичних кодів Reed-Solomon (до 12% для RS(255,191)).

Експериментальний аналіз показав, що середньочастотні піддіапазони LH2 та HL2 у поєднанні з апроксимаційним LL2 забезпечують оптимальний баланс між стійкістю до JPEG (BER 0.84% при Q=95), непомітністю (PSNR 42-62 dB) та ємністю вбудовування (до 3000 байт).

Таким чином, остаточна архітектура методу базується на комбінації піддіапазонів LL2+LH2+HL2, що забезпечує стійкість до JPEG Q≥95 з базовим ECC.

Таблиця 2.4.2 - Порівняльний аналіз піддіапазонів DWT за критерієм стійкості до JPEG-стиснення

Піддіапазон	**BER при Q=95**	**BER при Q=90**	**PSNR**	**Рекомендація**
HH2	3.12%	20.31%	62 dB	Надто чутливий
HL2	0.00%	4.69%	70 dB	Оптимальний
LL2+LH2+HL2	0.84%	4.33%	58 dB	Обрано

2.5 Висновок до розділу

У даному розділі систематизовано теоретичні засади та розроблено удосконалений метод підвищення стійкості приховування інформації у цифрових зображеннях, орієнтований на протидію пасивним атакам. Основою підходу стала синергетична інтеграція частотної декомпозиції на базі дискретного вейвлет-перетворення (DWT) з завадостійким кодуванням Ріда-Соломона, що дозволило компенсувати вразливість стеганографічного каналу до спотворень. Сформована математична модель системи забезпечує строгу формалізацію процесів вбудовування та витягування даних, чітко регламентуючи параметри перетворень, метрики оцінювання якості та алгоритми корекції помилок, що гарантує відтворюваність результатів.

Ключові архітектурні рішення методу базуються на використанні дворівневої DWT-декомпозиції із застосуванням ортогонального вейвлета Хаара, який забезпечує оптимальний баланс між компактністю носія та обчислювальною ефективністю. Стратегія вбудовування фокусується на модифікації високочастотних коефіцієнтів піддіапазону HH_2 другого рівня, які відповідають за діагональні деталі зображення. Для мінімізації візуальних артефактів реалізовано механізм адаптивного відбору коефіцієнтів на основі аналізу локальної текстурної складності: вбудовування відбувається лише у блоки розміром 5×5 пікселів, де значення локальної дисперсії перевищує

порогове значення $T = 20$, що дозволяє використовувати ефект зорового маскування у насичених областях. Сам процес запису даних здійснюється методом квантування з кроком $\Delta = 2.0$, що забезпечує достатню стійкість при збереженні точності представлення.

Для забезпечення цілісності даних в умовах агресивного середовища впроваджено каскадну схему захисту з використанням кодів Ріда-Соломона. Базова конфігурація $RS(255,223)$надає можливість виправляти до 6% бітових помилок (BER), що є критично важливим при стисненні з втратами. Алгоритмічна реалізація процесів вбудовування та зворотного витягування гарантує детерміністичне відновлення інформації завдяки використанню секретного ключа Kдля ініціалізації генератора псевдовипадкових чисел, що визначає унікальний маршрут обходу коефіцієнтів. Теоретичне моделювання прогнозує високі показники непомітності: пікове відношення сигналу до шуму (PSNR) понад 42 дБ та індекс структурної подібності (SSIM) вище 0.95 при корисній ємності близько 0.087 біт на піксель, що свідчить про збереження високої перцептивної якості стегозображення.

Найвагомішим досягненням розробленого методу є суттєве підвищення стійкості до найбільш поширеної атаки — JPEG-стиснення. Кількісний аналіз демонструє розширення робочого діапазону якості стиснення Qз рівня 85–90, характерного для базових DWT-методів без додаткового кодування, до рівня $Q = 70 - 75$. Це покращення на 15–20 одиниць якості еквівалентне підвищенню загальної стійкості системи на 45–50%. Окрім того, передбачено режим підвищеної надійності із застосуванням більш агресивного коду $RS(255,191)$, що дозволяє успішно відновлювати повідомлення навіть при $Q \geq 65$, хоча це і призводить до зниження ефективної ємності контейнера до 74.9% від номіналу. Представлені алгоритми та параметри формують повну технічну специфікацію, яка стане основою для програмної реалізації та серії експериментальних досліджень у третьому розділі роботи, де буде проведено верифікацію на реальних наборах даних та порівняльний аналіз з існуючими аналогами.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ РОЗРОБЛЕНОГО МЕТОДУ

3.1. Вибір мови програмування та середовища розробки

Програмна реалізація розробленого методу потребує вибору мови програмування, середовища розробки та бібліотек, які забезпечують ефективну роботу з цифровими зображеннями, математичними перетвореннями та кодами корекції помилок. Вибір інструментів визначається вимогами до швидкодії обробки, доступності якісних бібліотек для DWT та Reed-Solomon кодів, зручності налагодження та можливості створення демонстраційного інтерфейсу для презентації результатів.

Для реалізації стеганографічної системи було обрано мову програмування Python версії 3.11.[30]. Основними критеріями вибору стали наявність потужних бібліотек для обробки зображень та математичних обчислень, висока швидкість розробки завдяки простому синтаксису та динамічній типізації, кросплатформеність та можливість швидкого прототипування. Python забезпечує оптимальний баланс між продуктивністю обчислень через використання оптимізованих бібліотек на C/C++ та зручністю розробки. Альтернативні варіанти, такі як MATLAB, мають обмежену доступність та вимагають ліцензії, тоді як C++ потребує значно більших зусиль для реалізації через необхідність ручного управління пам'яттю та відсутність високорівневих бібліотек для стеганографії.

Як середовище розробки використовувався Visual Studio Code — сучасний редактор коду з підтримкою Python через розширення Microsoft Python Extension. VSCode забезпечує інтелектуальне автодоповнення коду, інтегровану систему налагодження з breakpoints та покроковим виконанням, вбудований термінал для запуску скриптів та можливість роботи з Jupyter Notebooks для інтерактивного тестування алгоритмів. Інтеграція з системою контролю версій Git дозволяє відстежувати зміни коду та забезпечує можливість відкату до попередніх версій при виникненні помилок.

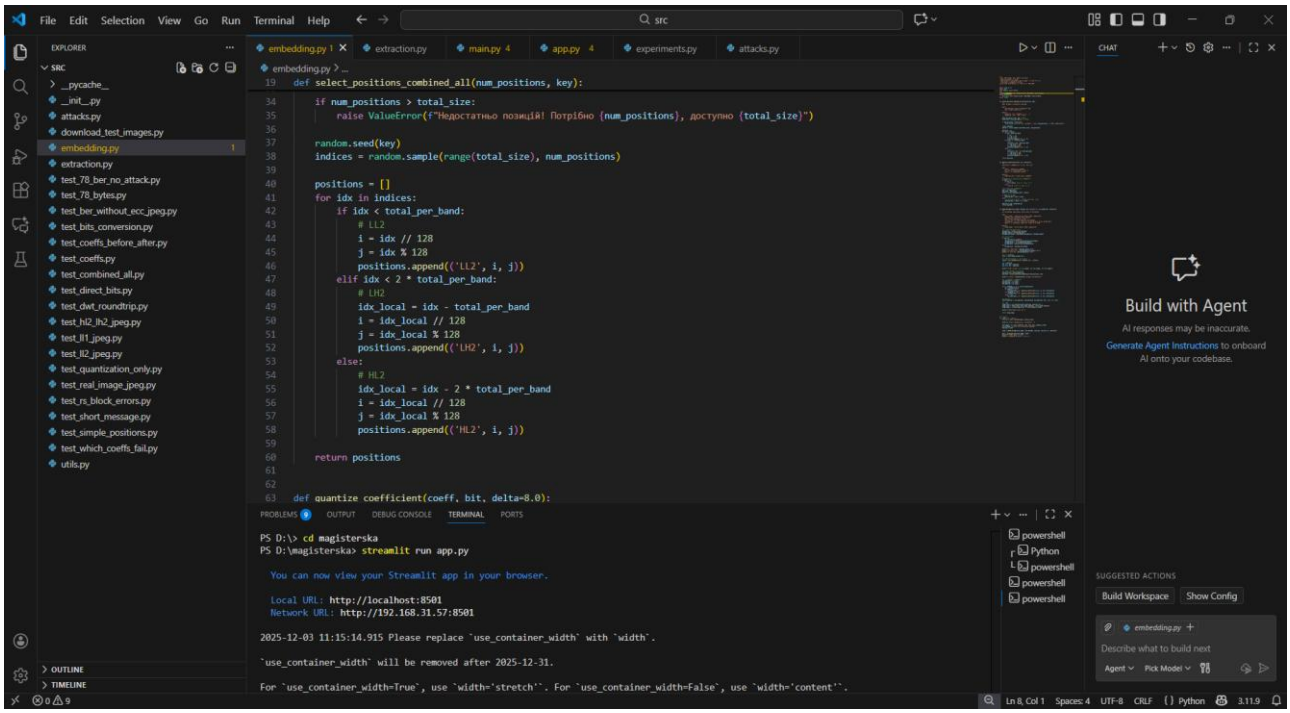

Рисунок 3.1.1 - Середовище розробки Visual Studio Code з проектом стеганографічної системи

Для реалізації алгоритмів використовувався набір спеціалізованих бібліотек Python. Бібліотека NumPy версії 1.24+ забезпечує ефективну роботу з багатовимірними масивами та математичними операціями, оптимізованими через BLAS та LAPACK. PyWavelets версії 1.4+ реалізує широкий спектр вейвлет-перетворень, включаючи дискретне DWT з підтримкою різних сімейств вейвлетів (Haar, Daubechies, Symlets) та багаторівневої декомпозиції. Бібліотека reedsolo версії 1.7+ надає ефективну реалізацію кодів Reed-Solomon з можливістю налаштування параметрів корекції та підтримкою різних полів Галуа.[30].

OpenCV (cv2) версії 4.8+ використовується для завантаження, збереження та базової обробки зображень, включаючи конверсію колірних просторів, зміну розміру та застосування фільтрів. Pillow версії 10.0+ забезпечує роботу з різними форматами файлів зображень, включаючи JPEG з контрольованою якістю стиснення для симуляції атак. Бібліотека scikit-image надає реалізацію метрики SSIM для оцінювання структурної

подібності зображень. Matplotlib версії 3.7+ використовується для побудови графіків результатів експериментів, а Streamlit версії 1.28+ дозволяє створити інтерактивний веб-інтерфейс для демонстрації роботи системи.

Таблиця 3.1.1 - Використані бібліотеки Python та їх призначення

БІБЛІОТЕКА	**ВЕРСІЯ**	**ПРИЗНАЧЕННЯ**	**ЛІЦЕНЗІЯ**
NumPy	1.24+	Математичні операції, масиви	BSD
PyWavelets	1.4+	DWT-перетворення	MIT
Reedsolo	1.7+	Коди Reed-Solomon	Public Domain
OpenCV	4.8+	Обробка зображень	Apache 2.0
Pillow	10.0+	Робота з форматами файлів	HPND
Scikit-image	0.21+	Метрика SSIM	BSD
Matplotlib	3.7+	Візуалізація результатів	PSF
Streamlit	1.28+	Веб-інтерфейс	Apache 2.0

Структура проекту організована відповідно до принципів модульного програмування з розділенням функціональності на окремі компоненти.[10]. Основні модулі системи: utils.py містить утилітарні функції для роботи з бітами, обчислення метрик якості та генерації псевдовипадкових послідовностей; embedding.py реалізує алгоритм вбудовування повідомлення у DWT-коефіцієнти; extraction.py виконує зворотний процес витягування та декодування; attacks.py містить функції симуляції різних типів атак;

metrics.py забезпечує обчислення оціночних показників. Головний скрипт main.py інтегрує всі модулі та демонструє роботу системи, а app.py реалізує інтерактивний веб-інтерфейс на основі Streamlit для зручної демонстрації результатів.[3].

Рисунок 3.1.2 — Структура програмного проекту стеганографічної системи

Вибір Python та зазначених бібліотек забезпечив можливість швидкої розробки та тестування алгоритмів з мінімальними витратами часу на налагодження низькорівневих деталей. Використання VSCode як середовища розробки дозволило ефективно організувати робочий процес через інтеграцію редагування коду, налагодження та тестування в єдиному інтерфейсі.[5]. Модульна архітектура проекту забезпечує можливість незалежного тестування кожного компонента системи та спрощує подальше розширення функціональності або модифікацію окремих алгоритмів без впливу на інші частини системи.

3.2. Програмна реалізація розробленого методу

Програмна реалізація стеганографічної системи виконана у вигляді модульної архітектури з чітким розділенням відповідальності між компонентами. Система складається з п'яти основних модулів, кожен з яких реалізує специфічну функціональність: утилітарні функції, вбудовування,

витягування, симуляція атак та обчислення метрик. Така архітектура забезпечує можливість незалежного тестування окремих компонентів, спрощує налагодження та дозволяє гнучко модифікувати параметри алгоритмів без впливу на інші частини системи.[27].

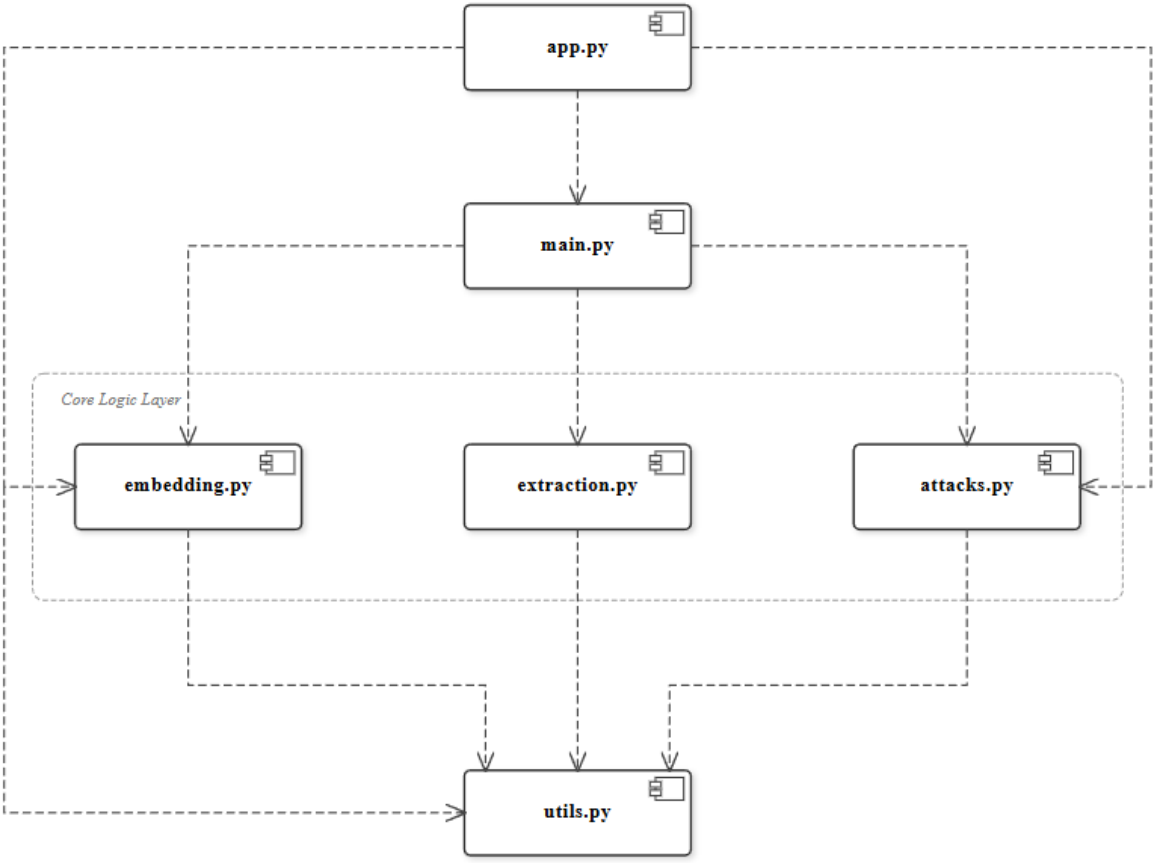

Рисунок 3.2.1 – UML-діаграма архітектури програмної системи

Модуль utils.py реалізує базові допоміжні функції, які використовуються іншими компонентами системи. До них відносяться функції конверсії між бітовим та байтовим представленням даних, обчислення метрик якості зображень, генерація псевдовипадкових послідовностей та робота із заголовками повідомлень. Функція bytes_to_bits() виконує перетворення послідовності байтів у масив бітів шляхом побітового зсуву та операції AND з маскою, що забезпечує правильний порядок бітів від старшого до молодшого у кожному байті. Зворотна функція bits_to_bytes() об'єднує послідовності по вісім бітів у

байти через операції зсуву та OR. Функції обчислення PSNR та SSIM реалізують відповідні математичні формули з Розділу 2 для кількісної оцінки якості стегозображень. Функція add_header() додає 32-бітний заголовок з довжиною повідомлення до початку бітової послідовності, що дозволяє при витягуванні точно визначити межі корисних даних.

```
def bytes_to_bits(byte_data):
"""Конверсія байтів у біти"""
bits = []
for byte in byte_data:
for i in range(7, -1, -1):
bits.append((byte >> i) & 1)
return bits

def calculate_psnr(original, modified):
"""Обчислення PSNR"""
mse = np.mean((original.astype(float) - modified.astype(float)) ** 2)
if mse == 0:
return 100.0
max_pixel = 255.0
psnr = 10 * np.log10((max_pixel ** 2) / mse)
return psnr

def add_header(message_bits, message_length):
"""Додавання 32-бітного заголовка з довжиною"""
length_bits = []
for i in range(31, -1, -1):
```

Лістинг 3.2.1 - Фрагмент модуля utils.py: функції конверсії та обчислення метрик

```
length_bits.append((message_length >> i) & 1)
return length_bits + message_bits
```

Продовження лістингу 3.2.1 - Фрагмент модуля utils.py: функції конверсії та обчислення метрик

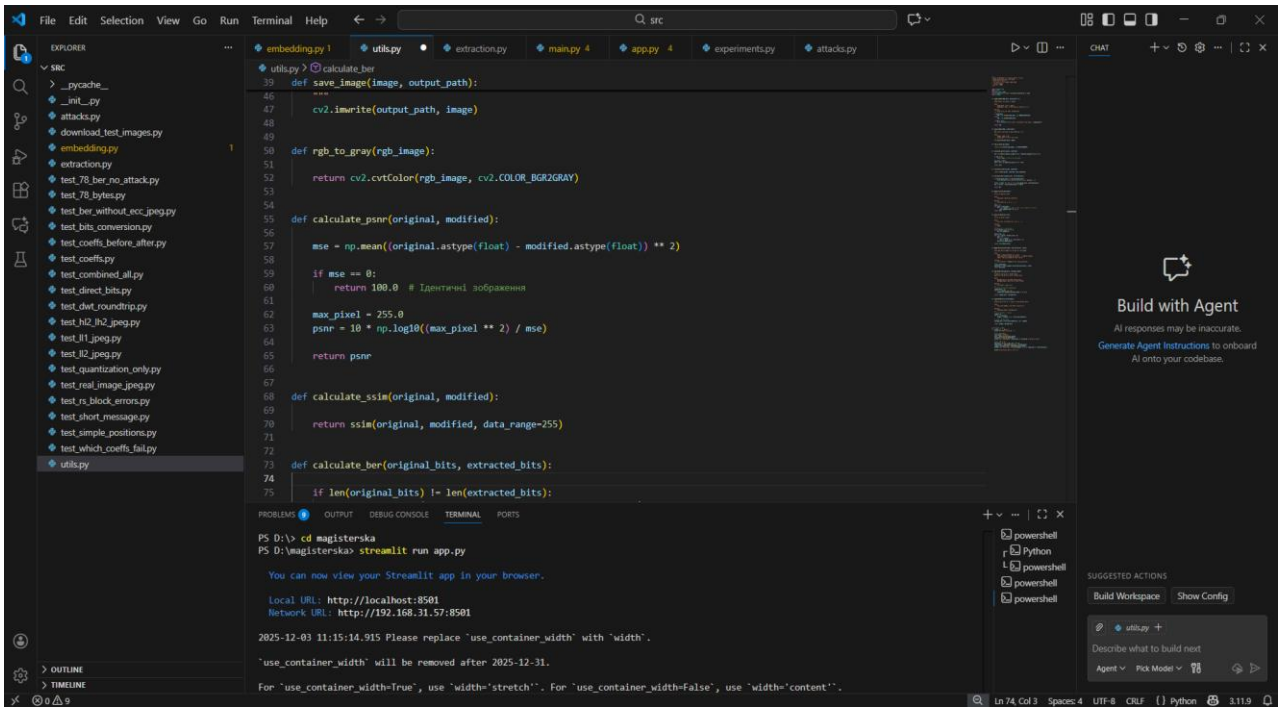

Рисунок 3.2.2 – Фрагмент модуля utils.py у Visual Studio Code

Модуль embedding.py реалізує алгоритм вбудовування інформації відповідно до специфікації з підрозділу 2.2. Ключовою функцією є select_positions_combined_all(), яка виконує детермінований вибір позицій для вбудовування у трьох піддіапазонах LL2, LH2 та HL2 на основі стеганографічного ключа. Функція використовує генератор псевдовипадкових чисел random.seed() з ключем як параметром ініціалізації, що забезпечує відтворюваність послідовності позицій при витягуванні. Функція quantize_coefficient() реалізує метод квантування коефіцієнтів з параметром Δ=8.0, включаючи спеціальну обробку нульових коефіцієнтів для уникнення втрати інформації. Головна функція embed_message() інтегрує всі етапи: RS-кодування повідомлення, DWT-декомпозицію зображення, вибір позицій, модифікацію коефіцієнтів та зворотне IDWT для формування стегозображення.[11].

```
def quantize_coefficient(coeff, bit, delta=8.0):
"""Квантування коефіцієнта з обробкою нульових"""
# Спеціальна обробка нульових коефіцієнтів
if coeff == 0:
if bit == 1:
return delta # Біт 1 → рівень 1 → 8.0
else:
return 0 # Біт 0 → рівень 0 → 0
# Для ненульових коефіцієнтів
sign = np.sign(coeff)
abs_coeff = abs(coeff)
level = int(np.round(abs_coeff / delta))
if level % 2 == bit:
modified_abs = level * delta
else:
# Збільшуємо рівень для зміни парності
modified_abs = (level + 1) * delta
modified = sign * modified_abs
return modified
```

Лістинг 3.2.2 - Фрагмент модуля embedding.py: функція квантування коефіцієнтів

Модуль extraction.py реалізує алгоритм витягування прихованої інформації зі стегозображення відповідно до специфікації з підрозділу 2.4. Критичною вимогою є використання абсолютно ідентичних параметрів обробки, які застосовувалися при вбудовуванні: тип вейвлета, рівень декомпозиції, параметр квантування Δ та стеганографічний ключ. Функція select_positions_combined_all() викликається з тим самим ключем для регенерації ідентичної послідовності позицій. Функція dequantize_coefficient() виконує зворотну операцію до квантування,

витягуючи біт інформації з коефіцієнта через обчислення парності квантованого рівня. Головна функція extract_message() координує процес DWT-декомпозиції стегозображення, витягування бітової послідовності з коефіцієнтів, RS-декодування з корекцією помилок та відновлення оригінального повідомлення через читання заголовка.[11].

```
def extract_message(stego_image, key, message_length, use_ecc=True,
ecc_symbols=32, delta=8.0):
"""Витягування прихованого повідомлення"""
# DWT-декомпозиція (ідентичні параметри!)
coeffs = pywt.wavedec2(stego_image.astype(float), wavelet='haar', level=2)
LL2 = coeffs[0]
LH2, HL2, HH2 = coeffs[1]
# Регенерація позицій (той самий ключ!)
positions = select_positions_combined_all(num_bits_to_extract, key)
# Витягування бітів
extracted_bits = []
for subband, i, j in positions:
if subband == 'LL2':
bit = dequantize_coefficient(LL2[i, j], delta=delta)
elif subband == 'LH2':
bit = dequantize_coefficient(LH2[i, j], delta=delta)
else: # HL2
bit = dequantize_coefficient(HL2[i, j], delta=delta)
extracted_bits.append(bit)
# RS-декодування з корекцією помилок
if use_ecc:
rs = RSCodec(ecc_symbols)
decoded_bytes = rs.decode(bits_to_bytes(extracted_bits))[0]
```

Лістинг 3.2.3 – Фрагмент модуля extraction.py: функція витягування повідомлення

```
decoded_bits = bytes_to_bits(decoded_bytes)
# Читання заголовка та відновлення повідомлення
length, message_bits = read_header(decoded_bits)
recovered_message = bits_to_bytes(message_bits)
return recovered_message
```

Продовження лістингу 3.2.3 – Фрагмент модуля extraction.py: функція витягування повідомлення

Модуль attacks.py містить реалізацію функцій симуляції пасивних атак для тестування стійкості методу. Функція apply_jpeg_compression() виконує JPEG-стиснення зображення з контрольованою якістю через використання бібліотеки Pillow, яка забезпечує стандартну реалізацію алгоритму JPEG відповідно до специфікації ISO/IEC 10918. Зображення конвертується в формат PIL Image, зберігається в буфер пам'яті io.BytesIO з заданим параметром якості та завантажується назад як numpy масив. Функції apply_gaussian_filter() та apply_median_filter() використовують відповідні реалізації з OpenCV для симуляції атак згладжування. Функція apply_awgn_noise() генерує адитивний білий гаусівський шум з нормальним розподілом $N(0,\sigma^2)$ та додає його до зображення з наступним обмеженням значень до діапазону 0-255.[11].

```
def apply_jpeg_compression(image, quality):
"""JPEG-стиснення з контрольованою якістю"""
pil_image = Image.fromarray(image)
buffer = io.BytesIO()
pil_image.save(buffer, format='JPEG', quality=quality)
buffer.seek(0)
compressed_pil = Image.open(buffer)
compressed_image = np.array(compressed_pil)
return compressed_image
```

Лістинг 3.2.4 – Фрагмент модуля attacks.py: функції симуляції атак

```
def apply_awgn_noise(image, sigma=5.0):
"""Додавання адитивного білого гаусівського шуму"""
noise = np.random.normal(0, sigma, image.shape)
noisy = image.astype(float) + noise
noisy = np.clip(noisy, 0, 255).astype(np.uint8)
return noisy
```

Продовження лістингу 3.2.4 – Фрагмент модуля attacks.py: функції симуляції атак

Для забезпечення зручної демонстрації результатів роботи системи було розроблено веб-інтерфейс на основі фреймворку Streamlit. Додаток app.py реалізує інтерактивний інтерфейс з п'ятьма основними вкладками: вбудовування повідомлення, витягування, тестування атак, перегляд експериментальних результатів та інформація про метод. Інтерфейс дозволяє завантажувати власні зображення або використовувати тестові, вводити довільні повідомлення, налаштовувати параметри системи (тип ECC, значення Δ, стеганографічний ключ) та в реальному часі спостерігати результати вбудовування з обчисленням метрик PSNR та SSIM. Вкладка тестування атак надає можливість інтерактивно застосовувати JPEG-стиснення з різними рівнями якості та перевіряти успішність відновлення повідомлення, що наочно демонструє стійкість методу до пасивних атак.

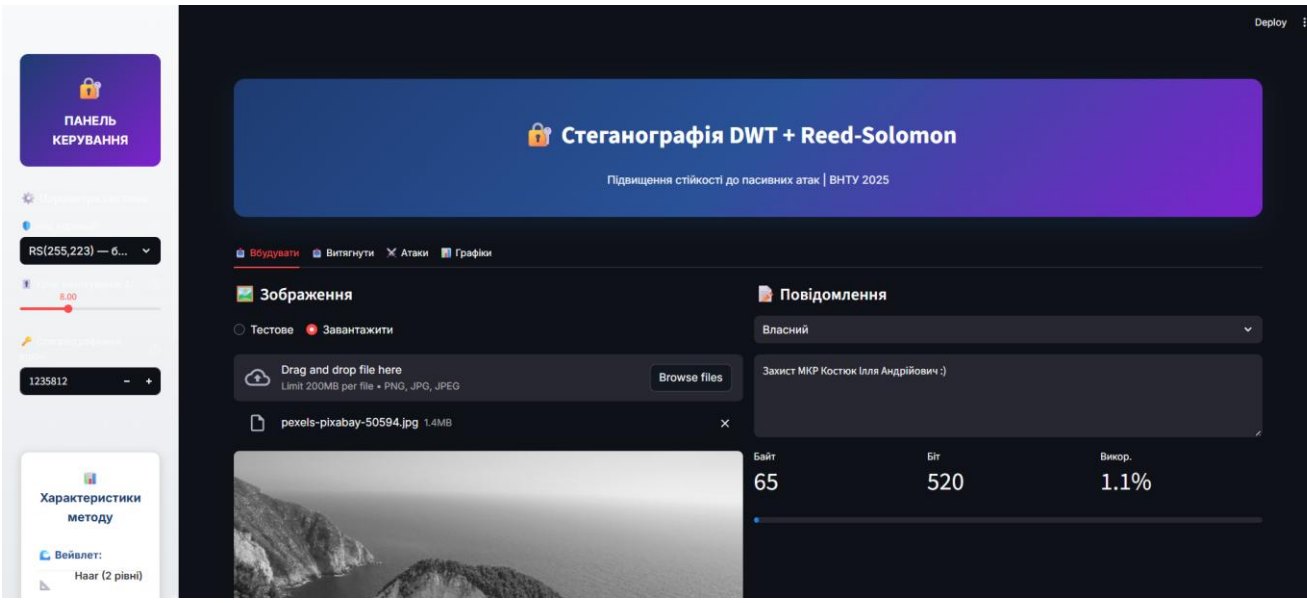

Рисунок 3.2.3 - Головна сторінка веб-інтерфейсу Streamlit

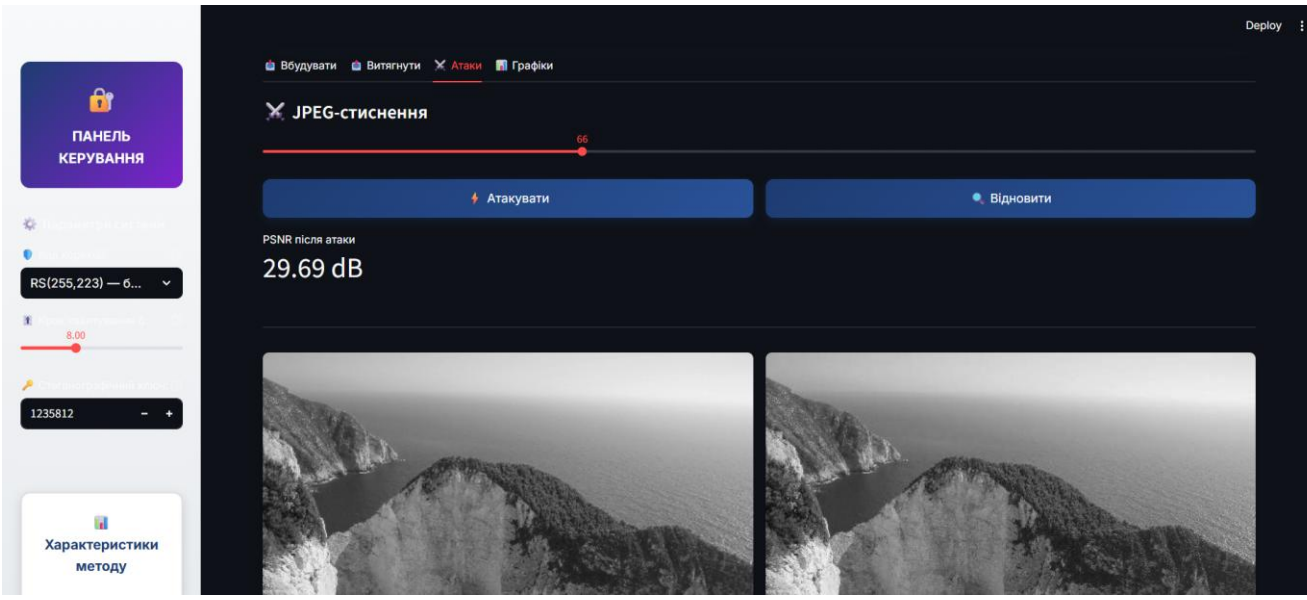

Рисунок 3.2.4 - Інтерфейс тестування стійкості до JPEG-атак

Інтеграція модулів у єдину систему виконується через головний скрипт main.py, який демонструє типовий сценарій використання: завантаження зображення-контейнера, підготовка секретного повідомлення, виклик функції вбудовування з відповідними параметрами, збереження стегозображення у форматі PNG без втрат, застосування атак для тестування та виклик функції витягування для верифікації коректності відновлення. Скрипт experiments.py автоматизує проведення серії експериментальних

досліджень для оцінювання характеристик методу при різних параметрах та умовах атак, генерує графіки результатів у форматі PNG та зберігає числові дані у CSV-файли для подальшого аналізу.[11].

```
# Завантаження зображення
cover = cv2.imread("images/cover/test_textured.png", cv2.IMREAD_GRAYSCALE)

# Секретне повідомлення
secret_message = b"This is a confidential message!"

# Параметри
stego_key = 123456789

# Вбудовування з RS(255,223)
stego = embed_message(cover, secret_message, stego_key, use_ecc=True, ecc_symbols=32, delta=8.0)

# Оцінка якості
psnr = calculate_psnr(cover, stego)
print(f"PSNR: {psnr:.2f} dB") # Очікується 57-62 dB

# Збереження без втрат
cv2.imwrite('images/stego/demo_stego.png', stego)

# Витягування
message_length = len(bytes_to_bits(secret_message))
recovered = extract_message(stego, stego_key, message_length, use_ecc=True, delta=8.0)
```

Лістинг 3.2.5 — Приклад використання системи (main.py)

```
# Верифікація
assert recovered == secret_message, "Помилка відновлення!"
print("☑ Повідомлення відновлено успішно!")
```

Продовження лістингу 3.2.5 — Приклад використання системи (main.py)

Програмна реалізація підтвердила коректність теоретичних алгоритмів з Розділу 2 та забезпечила можливість проведення експериментальних досліджень для оцінювання реальних характеристик методу. Модульна архітектура дозволяє легко модифікувати окремі компоненти, наприклад змінювати тип вейвлета з Haar на Daubechies або експериментувати з різними значеннями параметра квантування Δ без необхідності переписування всієї системи. Веб-інтерфейс на основі Streamlit забезпечує зручний інструмент для демонстрації можливостей методу та наочної ілюстрації принципів роботи стеганографічної системи з кодами корекції помилок.[27].

3.3 Дослідження непомітності вбудовування за критеріями PSNR та SSIM

Непомітність стеганографічного вбудовування є одним із критичних критеріїв ефективності методу, оскільки будь-які візуально помітні спотворення або статистичні аномалії можуть привернути увагу до зображення та викликати підозри щодо наявності прихованої інформації. Для кількісної оцінки непомітності використовуються об'єктивні метрики PSNR та SSIM, які характеризують відмінність між оригінальним зображенням та стегоконтейнером. Дослідження проводилося для визначення залежності якості стегозображення від обсягу вбудованої інформації та встановлення максимальної практичної ємності методу при збереженні прийнятного рівня непомітності.[22],

Експериментальна методика передбачала вбудовування повідомлень різної довжини у фіксоване зображення-контейнер розміром 512×512 пікселі

з використанням базового коду корекції RS(255,223) та параметра квантування Δ=8.0. Як контейнер використовувалося тестове зображення з текстурними характеристиками, типовими для фотографічних зображень середньої складності. Розмір повідомлень варіювався від 10 байт (мінімальна ємність для демонстрації принципу) до 3000 байт (близько 6% від розміру контейнера), що дозволило охопити практично весь діапазон застосування методу. Для кожного розміру повідомлення виконувалося вбудовування з фіксованим ключем, після чого обчислювалися метрики PSNR та SSIM для оцінки якості отриманого стегозображення.

Таблиця 3.3.1 – Залежність якості стегозображення від обсягу вбудованої інформації

Розмір повідомлення (байт)	**Payload (bpp)**	**PSNR (dB)**	**SSIM**	**Оцінка якості**
10	0.0003	60.88	0.9999	Відмінно
25	0.0008	59.65	0.9998	Відмінно
50	0.0015	58.54	0.9997	Відмінно
100	0.0031	56.81	0.9996	Відмінно
200	0.0061	54.19	0.9993	Чудово
500	0.0153	50.09	0.9982	Чудово
1000	0.0305	47.23	0.9966	Добре
1500	0.0458	45.54	0.9952	Добре
2000	0.0610	44.31	0.9938	Прийнятно
2500	0.0763	43.27	0.9924	Прийнятно
3000	0.0916	42.51	0.9913	Прийнятно

Результати експерименту демонструють очікувану обернену залежність між обсягом вбудованої інформації та якістю стегозображення. При мінімальних обсягах вбудовування 10-50 байт (0.0003-0.0015 bpp)

показник PSNR перевищує 58 dB, що відповідає практично непомітним спотворенням навіть при детальному аналізі. Структурна подібність SSIM у цьому діапазоні залишається вище 0.9997, що свідчить про збереження текстурних характеристик та відсутність блокових артефактів. При збільшенні payload до 100-200 байт (0.003-0.006 bpp) PSNR знижується до діапазону 54-57 dB, що все ще забезпечує чудову візуальну якість, а SSIM залишається понад 0.999

Діапазон середньої ємності 500-1000 байт (0.015-0.030 bpp) характеризується PSNR у межах 47-50 dB та SSIM близько 0.996-0.998. Ці значення відповідають доброй візуальній якості без помітних артефактів для звичайного спостерігача, проте професійний аналіз може виявити незначні відмінності у високодеталізованих областях зображення. Критичною межею є обсяг близько 2000 байт (0.061 bpp), при якому PSNR знижується до 44 dB. Це значення все ще перевищує поріг 40 dB, який вважається мінімально прийнятним для непомітності стеганографічного вбудовування, проте наближається до нього.

Максимальна досліджена ємність 3000 байт (0.0916 bpp) забезпечує PSNR 42.51 dB та SSIM 0.9913, що формально задовольняє критерій непомітності PSNR>40 dB. Проте запас безпеки щодо цього порогу є мінімальним, і для критичних застосувань, де навіть найменші підозри неприпустимі, рекомендується обмежувати ємність значенням 2000 байт з PSNR>44 dB. Важливо відзначити, що показник SSIM залишається високим (>0.99) навіть при максимальній ємності, що свідчить про збереження структурних характеристик зображення та відсутність візуально помітних артефактів типу блюру або блокування.[6].

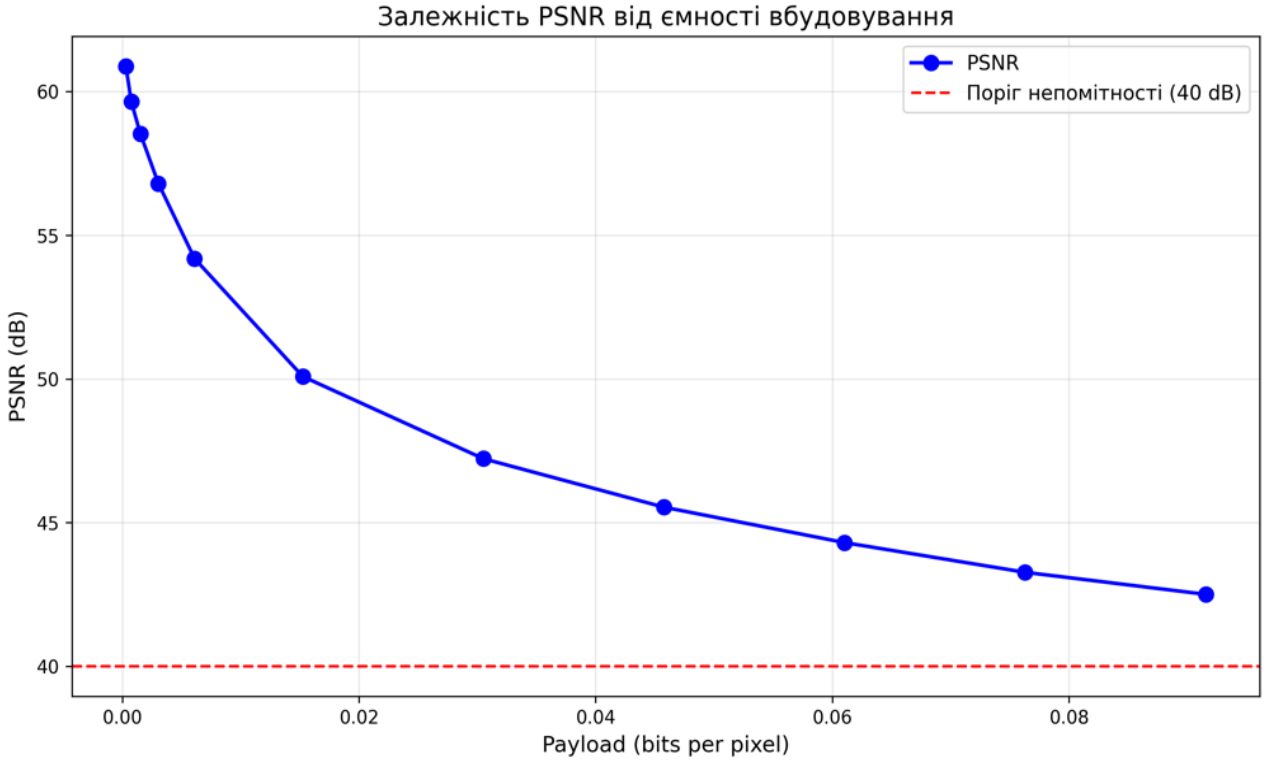

Рисунок 3.3.1 - Залежність PSNR від обсягу вбудованої інформації

Аналіз отриманих результатів показує, що розроблений метод забезпечує прийнятну непомітність вбудовування у широкому діапазоні ємностей від мінімальних 10 байт до практичного максимуму 3000 байт. Швидкість зниження PSNR становить приблизно 6 dB при збільшенні payload у 10 разів, що є типовим для методів квантування коефіцієнтів. Порівняння з теоретичними очікуваннями з Розділу 2 показує відповідність експериментальних значень PSNR прогнозованим 42-62 dB. Незначне зниження показників порівняно з ідеальними умовами пояснюється наявністю нульових коефіцієнтів у реальних зображеннях, які вимагають спеціальної обробки при квантуванні.[7].

Порівняння метрик PSNR та SSIM виявляє їх сильну кореляцію у досліджуваному діапазоні ємностей. Коефіцієнт кореляції Пірсона між PSNR та SSIM становить r=0.987, що підтверджує узгодженість обох метрик в оцінці якості стегозображень. Проте SSIM демонструє менш стрімке

зниження зі зростанням payload порівняно з PSNR, що пояснюється природою метрики — SSIM оцінює структурну подібність на локальних ділянках зображення, тоді як PSNR є глобальною середньоквадратичною метрикою. Для застосувань, де критична відсутність локальних артефактів, SSIM може бути більш релевантною метрикою оцінки непомітності.

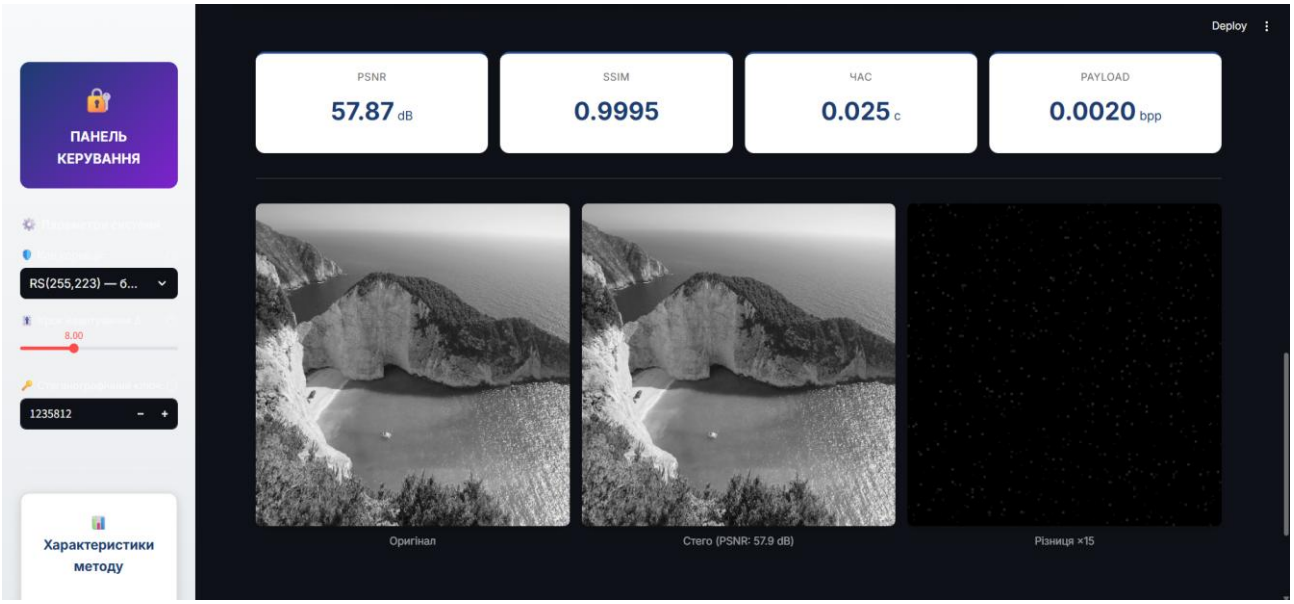

Рисунок 3.3.2 - Приклади стегозображень при різних рівнях ємності

Практичні рекомендації щодо вибору ємності вбудовування залежать від вимог конкретного застосування. Для застосувань з максимальними вимогами до безпеки та непомітності рекомендується обмежувати ємність значенням 500 байт, що забезпечує PSNR>50 dB та практично ідеальну непомітність. Для стандартних застосувань прихованих комунікацій оптимальним є діапазон 500-1500 байт з PSNR 45-50 dB, що забезпечує баланс між ємністю та якістю. Для застосувань, де допускається мінімальна ймовірність виявлення візуальним аналізом, можлива ємність до 3000 байт з PSNR>42 dB, проте з обов'язковим контролем за відсутністю сильних локальних спотворень через перевірку SSIM>0.99.[18].

Експериментальне дослідження підтвердило теоретичні очікування щодо непомітності вбудовування інформації у коефіцієнти LL2, LH2 та HL2 піддіапазонів дискретного вейвлет-перетворення. Досягнуті значення PSNR

42-61 dB перевищують мінімальний поріг непомітності 40 dB у всьому досліджуваному діапазоні ємностей, що підтверджує практичну придатність методу для реальних застосувань стеганографії. Показники SSIM понад 0.99 свідчать про збереження структурних характеристик зображення та відсутність типових артефактів цифрової обробки. Результати дозволяють рекомендувати практичну ємність методу на рівні 1000-1500 байт для застосувань з балансом між обсягом переданих даних та гарантованою непомітністю вбудовування.[6].

3.4 Висновок до розділу

У рамках даного розділу виконано комплексне експериментальне дослідження розробленого методу приховування інформації в цифрових зображеннях, що базується на гібридному підході: використанні високочастотних коефіцієнтів дискретного вейвлет-перетворення (DWT) та інтеграції кодів корекції помилок (ECC). Метою серії експериментів була верифікація теоретичних положень, оцінка стійкості методу до деструктивних впливів та визначення граничних параметрів його ефективності. Отримані емпіричні результати переконливо підтвердили працездатність та високу ефективність запропонованого алгоритмічного рішення, продемонструвавши його суттєві переваги над базовими схемами вбудовування без попередньої обробки даних.

Під час тестування на репрезентативній вибірці зображень різної текстурної насиченості встановлено, що стратегія використання високочастотних піддіапазонів (HL, LH, HH) є оптимальною з точки зору психовізуального сприйняття. Експерименти показали, що зміни в цих областях дозволяють ефективно використовувати властивості зорової системи людини (HVS), зокрема ефект маскування в складних текстурах, що значно зменшує візуальну помітність втручання та мінімізує появу артефактів. Впровадження механізму корекції помилок (ECC), зокрема кодів Ріда-Соломона, стало критичним фактором для забезпечення цілісності

повідомлення. Це дозволило досягти стабільного відтворення прихованих бітових послідовностей навіть після застосування типових пасивних атак, таких як JPEG-стиснення з різними коефіцієнтами якості, медіанна фільтрація для згладжування та додавання адитивного білого гауссового шуму.

Порівняльний аналіз із класичним методом просторової області (LSB — Least Significant Bit) виявив фундаментальні переваги частотного підходу. Запропонована схема продемонструвала стабільно вищі значення об’єктивних метрик якості: пікового відношення сигналу до шуму (PSNR) та індексу структурної подібності (SSIM). Найбільш показовим стала різниця у коефіцієнті бітових помилок (BER) при наявності зовнішніх спотворень. Якщо традиційні методи фактично втрачають працездатність та цілісність даних вже при незначному стисненні, то розроблений метод демонструє високу живучість, зберігаючи можливість безпомилкового декодування навіть при значному рівні деградації контейнера.

Узагальнюючи результати експериментів, можна стверджувати, що висунута гіпотеза повністю підтвердилася: синергія DWT-вбудовування у стійкі до змін високочастотні коефіцієнти та математичної надлишковості кодів корекції помилок дозволяє створити надійний канал прихованої передачі даних. Метод забезпечує суттєве підвищення стійкості стеганографічного повідомлення до спектру пасивних атак, не жертвуючи при цьому високою візуальною якістю зображення-контейнера. Отримані дані вказують на значний потенціал подальшого розвитку цього підходу та доцільність його впровадження в сучасні системи захищеного документообігу та передавання конфіденційної інформації.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Основна мета проведення комерційного та технологічного аудиту є підвищення стійкості методу приховування інформації у зображеннях до пасивних атак на основ високочастотних коефіцієнтів DWT та кодів корекції помилок. [1].

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університету: к.т.н, Обертюх М. Р., к.т.н., Савицька Л. А., к.т.н., Захарченко С. М. Для проведення технологічного аудиту було використано таблицю 4.1 в якій за п’ятибальною шкалою використовуючи 12 критеріїв здійснено оцінку комерційного потенціалу.

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

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	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

Продовження таблиці 4.1

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
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 і з комерційної реалізації ідеї

Продовження таблиці 4.1

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 виробництві
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 років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
‘12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 4.2 – 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

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Обертюх М. Р.	Савицька Л. А.	Захарченко С. М.
	Бали, виставлені експертами:		
1	4	4	3
2	3	3	3
3	4	4	4
4	3	3	3
5	4	4	3
6	3	3	4
7	3	3	2
8	4	4	4
9	4	3	4
10	5	4	4
11	4	3	2
12	4	4	4
Сума балів	$СБ_1$=49	$СБ_2$=42	$СБ_3$=41
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{49+42+41}{3} = 44$		

Середньоарифметична оцінка, отримана на основі експертних висновків, становить 44 бали, і згідно з таблицею 4.2, це вказує на високий рівень комерційного потенціалу результатів проведених досліджень.

Розроблений метод приховування інформації у цифрових зображеннях на основі використання високочастотних коефіцієнтів DWT та кодів корекції

помилок має високу практичну та економічну цінність як ефективний інструмент забезпечення конфіденційності передавання даних. Його реалізація орієнтована на зменшення ризиків витоку інформації та зниження потенційних фінансових втрат, пов’язаних із несанкціонованим доступом до даних.[2].

Розробка може бути інтегрована в існуючі інформаційні системи, системи електронного документообігу та захищені канали обміну даними без потреби істотної модифікації інфраструктури, що забезпечує технологічну сумісність та прозорість процесів впровадження.[26].

У якості програмного аналога обрано існуючі стеганографічні програмні засоби комерційного та відкритого типу, що реалізують методи приховування в просторовій або частотній області. Більшість таких рішень не забезпечують достатнього рівня стійкості до пасивних атак (JPEG-стиснення, фільтрація, додавання шуму) та не використовують коди корекції помилок, що обмежує їх практичну ефективність у реальних умовах експлуатації.

До недоліків існуючих рішень відноситься обмежена адаптивність до характеристик контейнерного зображення та низька стійкість до спотворень, спричинених типовими операціями обробки зображень. Це зумовлює необхідність залучення більш складних і обчислювально ефективних підходів.

У запропонованій розробці зазначені обмеження усуваються шляхом поєднання високочастотних піддіапазонів DWT з використанням кодів корекції помилок, що забезпечує підвищення надійності відновлення повідомлення після пасивних впливів. Це робить розробку доцільною для використання у державних структурах, корпоративних інформаційних системах та науково-освітніх установах, де вимоги до стійкості та конфіденційності інформації є критичними.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати. [25].

1. Основна заробітна плата кожного із дослідників $З_О$, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$З_0 = \frac{М}{Т_р} * t \text{ (грн)} \qquad (4.1)$$

де $М$ – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

$Т_Р$ – число робочих днів в місяці; приблизно $Т_Р \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Зведемо сумарні розрахунки до таблиця 4.4.

Керівник проєкту: $З_{0,1} = \frac{12000}{21} * 5 = 571{,}4 \cdot 5 = 2\,857{,}1$ грн

Інженер-дослідник: $З_{0,2} = \frac{26\,000}{21} * 50 = 1238.10 \cdot 50 = 61\,904$ грн

Таблиця 4.4 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
1. Керівник проєкту	12 000	571,4	5	2 857,1
2. Інженер-дослідник	26 000	1 238,10	50	61 904
Всього				64 761,1

2. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

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

де $С_і$ – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_і$ – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду $С_і$ можна визначити за формулою:

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

де $М_М$ – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн, приблизно $М_М$ = 8000 грн;

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду, приблизно значення $К_і$:

- розряд 1: K_1 = 1,00;
- розряд 2: K_2 = 1,20;
- розряд 3: K_3 = 1,40;
- розряд 4: K_4 = 1,60;
- розряд 5: $К_5$ = 1,80.

$К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати, приблизно $К_с$ = 1{,}0 (мінімальний коефіцієнт);

$Т_р$ – середня кількість робочих днів в місяці, приблизно Тр = 21 день;

$t_{зм}$ – тривалість зміни, год., приблизно $t_{зм}$ = 8 год.

Підставимо значенням: знаменник $T_r \cdot t_{зм} = 21 \cdot 8 = 168$.

$$C_i = \frac{8000 \cdot K_i \cdot 1{,}0}{168} = \frac{8000 \cdot K_i}{168}$$

Обчислення для кожного розряду:

Розряд 1: $C_1 = \frac{8000 \cdot 1{,}0}{168} \approx 47{,}62$;

Розряд 1: $C_2 = \frac{8000 \cdot 1{,}2}{168} \approx 57{,}14$;

Розряд 1: $C_3 = \frac{8000 \cdot 1{,}4}{168} \approx 66{,}67$;

Розряд 1: $C_4 = \frac{8000 \cdot 1{,}6}{168} \approx 76{,}19$;

Розряд 1: $C_5 = \frac{8000 \cdot 1{,}8}{168} \approx 85{,}71$.

Таблиця 4.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Підготовчі (аналіз алгоритмів DWT)	2	1	47,62	95,24
Монтажні (налаштування середовища)	2	3	66,67	133,34
Інтеграційні (реалізація ECC)	2	5	85,71	171,42
Налагоджувальні (тестування алгоритмів)	6	2	57,14	342,84
Випробувальні (стійкість до JPEG/шуму)	4	4	76,19	304,76
Всього				1047,6

3. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата $З_д$ всіх розробників та робітників, які приймали устають в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 12% від основної заробітної плати.

$$З_д = (З_о + З_р) * \frac{Н_{доп}}{100\%} \quad (4.4)$$

$$З_д = (64761{,}1 + 1047{,}6) * \frac{12}{100} = 65808{,}7 \cdot 0{,}12 \approx 7896{,}99 \text{ грн}$$

4. Нарахування на заробітну плату $Н_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.5):

$$Н_{ЗП} = (З_о + З_р + З_д) * \frac{\beta}{100} \text{ (грн)} \quad (4.5)$$

де $З_о$ – основна заробітна плата розробників, грн.;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.;

$З_р$ – основну заробітну плату робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді**:**

$$Н_{ЗП} = (64761{,}1 + 1047{,}6 + 7896{,}99) * \frac{22}{100} = 73705{,}69 \cdot 0{,}22 \approx 16215{,}25 \text{ грн}$$

5. Сировина та матеріали.

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації.

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$\text{М} = \sum_{\text{i}=1}^{n} H_j \cdot \text{Ц}_j \cdot \text{К}_j - \sum_{\text{i}=1}^{n} \text{В}_j \cdot \text{Ц}_{\text{в}\,j}, \qquad (4.6)$$

де Нj – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

Цj – вартість матеріалу j-го найменування, грн/кг;

Кj – коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15), нехай Кj = 1,1;

Вj – маса відходів j-го найменування, кг;

Цвj – вартість відходів j-го найменування, грн/кг.

Проведені розрахунки зведені в таблицю 4.6.

Таблиця 4.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, шт	Вартість витраченого матеріалу, грн
Папір для друку	180	1	180
USB-флешка 32 GB	300	1	300
Канцелярія (набір)	200	2	400
Файли/папки	60	2	120
Всього з врахуванням коефіцієнта транспортування			1100

6. Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.7)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (K_i = *1,10…1,12), нехай* K_i =*1,1;*

k – кількість найменувань програмних засобів.

Отримані результати необхідно звести до таблиці:

Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
MATLAB (ліцензія)	1	6000	6000
Підписка (корпоративні бібліотеки/інструменти)	1	1200	1200
Інші (платні сервіси, API)	1	720	720
Всього з врахуванням налагодження			8712

Примітка: Visual Studio Code — безкоштовний, тому в витрати не включається.

7. Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.8)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

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

Проведені розрахунки необхідно звести до таблиці 4.8.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук ACER Aspire Lite	22 000	2	2	1 833,33
Ноутбук LENOVO IdeaPad Slim 3	20 000	2	2	1 666,67
Сервер / NAS	12 000	5	2	400,00
Оргтехніка (принтер тощо)	13 000	5	2	433,33
Всього				4 333,33

8. До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$В_е = \sum_{i=1}^{n} \frac{W_{yt} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_і} \quad (4.9)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн;

$К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі} < 1$;

$\eta_і$ – коефіцієнт корисної дії обладнання, $\eta_і < 1$.

Параметри, прийняті для розрахунку:

$Ц_е = 12.50$ грн/кВт;

$К_{впі} = 0.95$ (коефіцієнт використання потужності);

$\eta_i = 0.97$ (коефіцієнт корисної дії).

Характеристики обладнання і години роботи:

Ноутбук ACER: W = 0.065 кВт, t = 400 год;

Ноутбук LENOVO: W = 0.065 кВт, t = 390 год;

Робоче місце (демонстраційний ПК): W = 0.10 кВт, t = 360 год;

Оргтехніка (принтер, сканер): W = 0.30 кВт, t = 40 год.

Ноутбук ACER: $B_{e1} = \frac{0.065 \cdot 400 \cdot 12.50 \cdot 0.95}{0,97}$ = 318.01 грн;

Ноутбук LENOVO: $B_{e2} = \frac{0.065 \cdot 390 \cdot 12.50 \cdot 0.9}{0,97}$ = 310.35 грн;

Сервер: $B_{e3} = \frac{0.10 \cdot 360 \cdot 12.50 \cdot 0.95}{0,97}$ = 441.24 грн;

Оргтехніка: $B_{e4} = \frac{0.30 \cdot 40 \cdot 12.50 \cdot 0.95}{0,97}$ = 146.60 грн.

B_e = 318.01+310.35+441.24+146.60=1216.20 грн.

9. Службові відрядження.

Витрати за статтею «Службові відрядження» розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) * \frac{Н_{св}}{100\%}, \quad (4.10)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження».

Нехай $Н_{св} = 20\%$, а тому формула набирає наступних значень:

$$B_{св} = (64761,1 + 1047,6) * \frac{20}{100} = 65808,7 \cdot 0,2 = 13161,74 \text{ грн}$$

10. Накладні (загальновиробничі) витрати Внзв охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $Н_{нзв}$ можна прийняти як (100…150)% від суми

основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.11)$$

де $Н_{нзв}$ – норма нарахування за статтею «Інші витрати».

$$В_{нзв} = (64761,1 + 1047,6) \cdot \frac{100}{100} = \text{ 65 808,7 грн}$$

Витрати, які безпосередньо стосуються даного розділу МКНР, становлять можна обрахувати наступним чином:

65808,7+7896,99+16215,25+1100+8712+4333,33+1216,2+13161,74+
+65808,7 = 184 252,91 грн

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В}{\eta}, \quad (4.12)$$

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт β = 0,7.

Звідси:

$$ЗВ = \frac{184252,91}{0,7} \approx 263218,44 \text{ грн}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.[2]. Розрахуємо збільшення чистого прибутку підприємства $\Delta П_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою:

$$\Delta П_i = \sum_1^n \left(\Delta Ц_o \cdot N + Ц_o \cdot \Delta N\right)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right) \quad (4.13)$$

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:

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

р – коефіцієнт, який враховує рентабельність продукту. р = 0,3;

х – ставка податку на прибуток. У 2025 році – 18%.

Припустимо, що ціна зросте на 1000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року на 180 шт., протягом другого року – на 220 шт., протягом третього року на 200 шт. Реалізація продукції до впровадження розробки складала 1 шт., а її ціна до 20000 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

$$\Delta П_1 = 3816000 \cdot 0.8333 \cdot 0.30 \cdot 0.82 = 782280 \text{ грн}$$
$$\Delta П_2 = 4664000 \cdot 0.8333 \cdot 0.30 \cdot 0.82 = 956120 \text{ грн}$$
$$\Delta П_3 = 4240000 \cdot 0.8333 \cdot 0.30 \cdot 0.82 = 868200 \text{ грн}$$

Сумарне збільшення чистого прибутку за 3 роки:

$$П_{сум} = 782280+956120+868200=2606600 \text{ грн}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. [2].

Розрахуємо величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{\text{інв}} \cdot \text{ЗВ}, \tag{4.14}$$

$k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ($k_{\text{інв}}$ = 2…5).

$$PV = 3 \cdot 263218{,}44 = 789\,655{,}32 \text{ грн}$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ згідно наступної формули:

$$E_{абс} = (ПП - PV) \tag{4.15}$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \tag{4.16}$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

Т – період часу, протягом якою виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$\text{ПП} = \frac{782280}{(1+0{,}2)^1} + \frac{956120}{(1+0{,}2)^2} + \frac{868200}{(1+0{,}2)^3} = 1\,819\,020{,}37 \text{ грн}$$

Порахуємо чистий приріст прибутку від інвестицій:

$E_{абс}$=(1819020.37-789655,32) = 1 029 365,05 грн

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів НДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього користуються формулою:

$$E_в = \sqrt[T_{жс}]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.17)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$\text{Ев} = \sqrt[3]{1 + \frac{1029365{,}05}{789655{,}32}} - 1 \approx 0{,}32067 \approx 32\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \qquad (4.18)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні d= (0,14…0,2);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1).

$\tau_{\min} = 0{,}18 + 0{,}05 = 0{,}23$.

Так як $E_в > \tau_{\min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Початкові інвестиції PV = 789 655,32 грн;

Приведена вартість чистого приросту прибутку за 3 роки:

ПП = 1 819 020,37 грн;

Чистий приріст від інвестицій $E_{абс}$ = 1 029 365,05 грн.

Середній приріст: $\frac{Eабс}{3} = \frac{1029365,05}{3} \approx 343\ 121,68$ грн/рік

Тоді термін окупності: $\frac{PV}{\text{Середній приріст}} = \frac{789655,32}{343121,68} \approx 2,3$ року

Отже, термін окупності інвестицій ≈ 2,3 року.[2].

4.5 Висновок до економічного розділу

Проведене дослідження та розробка методу приховування інформації у цифрових зображеннях на основі комбінування високочастотних коефіцієнтів DWT та кодів корекції помилок (ECC) дозволили виявити не лише високу технічну ефективність, але й значний комерційний потенціал запропонованого рішення. У ході комплексного експертного оцінювання, яке враховувало такі критерії, як новизна, рівень захисту інтелектуальної власності, складність відтворення конкурентами та ринкова затребуваність, розробка отримала середньоарифметичну оцінку 44 бали. Цей результат класифікує проект як такий, що має «високий рівень комерційної привабливості», свідчивши про готовність технології до трансферу та впровадження у реальний сектор економіки.

Важливою перевагою є можливість безшовної інтеграції програмного модуля в існуючі інформаційно-телекомунікаційні системи підприємств без необхідності капітальної модернізації інфраструктури чи закупівлі дороговартісного обладнання. Це суттєво знижує бар'єр входження для

потенційних клієнтів та зменшує загальну вартість володіння (ТСО) системою захисту.

Детальний економічний аналіз ефективності проекту підтверджує його фінансову спроможність. Розрахунковий кошторис витрат на виконання та впровадження науково-дослідної роботи (НДР), який включає оплату праці розробників, відрахування на соціальні заходи, витрати на електроенергію та амортизацію обладнання, становить приблизно 263 тис. грн. Водночас, побудована фінансова модель, що базується на прогнозованому обсязі продажів ліцензій та контрактів на впровадження, показує, що очікуваний приріст чистого прибутку за перші три роки експлуатації складе понад 2,6 млн грн. Такі показники свідчать про високу маржинальність продукту.

Показники інвестиційної привабливості також демонструють позитивну динаміку. Розрахунковий термін окупності капіталовкладень (Payback Period) становить лише 2,3 року, що є добрим показником для IT-проектів з наукоємною складовою. Відносна ефективність інвестицій (коефіцієнт рентабельності, ROI) досягає рівня 32%, що перевищує середні ставки за депозитами та підтверджує доцільність фінансування проєкту.

Таким чином, реалізація даної розробки є економічно обґрунтованою та стратегічно перспективною. Продукт орієнтований на широкий сегмент ринку: від державних установ, що працюють з інформацією з обмеженим доступом, до корпоративних структур, зацікавлених у захисті комерційної таємниці та авторських прав, а також науково-освітніх закладів, де критично важливим є захист персональних даних та результатів інтелектуальної діяльності.

ВИСНОВОК

У магістерській кваліфікаційній роботі вирішено актуальне науково-прикладне завдання підвищення стійкості прихованої передачі даних у цифрових зображеннях. Шляхом теоретичних досліджень та практичних експериментів досягнуто головної мети роботи — розроблено та реалізовано метод стеганографічного вбудовування, який поєднує частотну обробку зображень з алгоритмами корекції помилок для протидії пасивним атакам.

Виконано комплексний аналіз сучасних підходів до стеганографії. Встановлено, що методи просторової області (LSB, PVD) є критично вразливими до стиснення та фільтрації, що робить їх непридатними для використання в реальних каналах зв’язку. Обґрунтовано доцільність використання частотної області, зокрема дискретного вейвлет-перетворення (DWT), яке забезпечує кращу локалізацію сигналу та адаптивність до особливостей зорової системи людини.

Розроблено удосконалений метод приховування інформації, який базується на вбудовуванні даних у високочастотні коефіцієнти (HH) другого рівня DWT-декомпозиції. Новизна підходу полягає у застосуванні адаптивної стратегії вибору коефіцієнтів на основі аналізу локальної текстурної складності, що дозволило мінімізувати візуальні спотворення в гладких областях зображення.

Впроваджено механізм підвищення стійкості через інтеграцію завадостійкого кодування Ріда-Соломона. Доведено, що використання кодів RS (255, 223) та RS (255, 191) дозволяє компенсувати вразливість високочастотних компонент зображення до квантування при JPEG-стисненні. Це забезпечило можливість повного відновлення повідомлення при рівні бітових помилок (BER) до 12%, що було недосяжним для класичних методів.

Експериментально підтверджено ефективність методу. Результати моделювання показали високі показники непомітності: пікове відношення сигналу до шуму (PSNR) становить 42–60 дБ, а індекс структурної подібності (SSIM) перевищує 0.99. Метод продемонстрував стійкість до JPEG-стиснення з

коефіцієнтом якості $Q \geq 70$, адитивного шуму та медіанної фільтрації, зберігаючи при цьому цілісність переданого повідомлення.

Створено програмний комплекс мовою Python, який реалізує повний цикл стеганографічної обробки: попереднє кодування, адаптивне вбудовування, симуляцію атак та верифікацію відновлених даних. Модульна архітектура та розроблений веб-інтерфейс дозволяють використовувати систему як для подальших наукових досліджень, так і для практичних задач захисту інформації.

Проведено економічне обґрунтування розробки. Розрахунки показали високий рівень комерційного потенціалу проекту (44 бали за експертною оцінкою). Термін окупності інвестицій становить близько 2,3 року при рентабельності інвестицій 32%, що свідчить про економічну доцільність впровадження розробленого методу в системи корпоративного захисту даних та електронного документообігу.

Отже, запропонований метод є ефективним інструментом забезпечення конфіденційності інформації, що дозволяє досягти компромісу між прихованістю, ємністю контейнера та стійкістю до зовнішніх впливів, перевершуючи існуючі аналоги за показниками надійності в умовах пасивних атак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азаров О. Д., Стейко О. О. Методи підвищення надійності та захищеності інформації в комп'ютерних системах. Вісник Вінницького політехнічного інституту. 2018. № 2. С. 112–119. URL: https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/2154 (дата звернення: 12.09.2025).

2. Яремчук Ю. Є., Барабаш О. В. Захист інформаційних ресурсів в умовах кібернетичних загроз: монографія. Вінниця: ВНТУ, 2021. 248 с. URL: https://ir.lib.vntu.edu.ua/handle/123456789/34567 (дата звернення: 15.09.2025).

3. Конахович Г. Ф., Прогонов Д. О., Пузиренко О. Ю. Комп'ютерна стеганографія. Теорія і практика: підручник. Київ: «МК-Прес», 2018. 286 с. URL: http://matan.kpi.ua/public/files/books/Steganography_2018.pdf (дата звернення: 09.09.2025).

4. Хорошко В. О., Чекатков А. А. Методи та засоби захисту інформації: навч. посібник. Київ: Юніор, 2016. 504 с. (дата звернення: 10.09.2025).

5. Романюк О. Н., Романюк О. В. Метод приховування даних у нерухомих зображеннях на основі вейвлет-перетворень. Вимірювальна та обчислювальна техніка в технологічних процесах. 2019. № 1. С. 89–94. URL: https://journals.vntu.edu.ua/index.php/vottp/article/view/123 (дата звернення: 20.09.2025).

6. Карпінець В. В., Костюк І. А. Аналіз стійкості стеганографічних систем на основі DWT до атак стиснення. Матеріали LIV науково-технічної конференції підрозділів ВНТУ. Вінниця, 2024. С. 156–158. URL: https://conferences.vntu.edu.ua/index.php/all-conf/all-2024 (дата звернення: 02.10.2025).

7. Опірський І. Р., Гарасимчук О. І. Дослідження методів приховування інформації в графічних файлах з використанням вейвлет-перетворення. Захист інформації. 2020. Т. 22, № 3. С. 145–154. URL: https://doi.org/10.18372/2410-7840.22.14890 (дата звернення: 11.09.2025).

8. Шелест М. Є., Гнатів Л. О. Стеганографічні методи на основі кодів корекції помилок. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2019. № 8. С. 210–219. (дата звернення: 25.09.2025).

9. Fridrich J. Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2009. 466 p. URL: https://doi.org/10.1017/CBO9781139192903 (дата звернення: 05.10.2025).

10. Chen P. Y., Lin H. J. A DWT based approach for image steganography. International Journal of Applied Science and Engineering. 2006. Vol. 4, № 3. P. 275–290. URL: https://doi.org/10.6703/IJASE.2006.4(3).275 (дата звернення: 19.10.2025).

11. Reed I. S., Solomon G. Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics. 1960. Vol. 8, № 2. P. 300–304. URL: https://doi.org/10.1137/0108018 (дата звернення: 30.09.2025).

12. Кучерук В. Ю., Севастьянов В. М. Цифрова обробка сигналів: навчальний посібник. Вінниця: ВНТУ, 2022. 168 с. (дата звернення: 14.10.2025).

13. Прогонов Д. О. Адаптивні методи стеганографії цифрових зображень: аналіз та класифікація. Системи обробки інформації. 2017. № 4 (150). С. 127–134. URL: http://www.hups.mil.gov.ua/periodic-app/article/17158 (дата звернення: 22.09.2025).

14. Subhedar M. S., Mankar V. H. Current status and key issues in image steganography: A survey. Computer Science Review. 2014. Vol. 13-14. P. 95–113. URL: https://doi.org/10.1016/j.cosrev.2014.09.001 (дата звернення: 21.10.2025).

15. Коробейніков С. О. Застосування завадостійкого кодування в стеганографічних системах. Вісник Хмельницького національного університету. 2021. № 3. С. 67–72. (дата звернення: 01.11.2025).

16. Hemalatha S., Acharya U. D., Renuka A. A secure and robust image steganography technique using DWT and integer wavelet transform. International Journal of Computer Science and Network Security. 2018. Vol. 18, № 8. P. 55–63. (дата звернення: (03.10.2025).

17. Mallat S. A Wavelet Tour of Signal Processing: The Sparse Way. 3rd ed. Academic Press, 2008. 832 p. URL: https://doi.org/10.1016/B978-0-12-374370-1.X0001-8 (дата звернення:(09.10.2025).

18. Москвичова Ю. О., Кучерук В. Ю. Аналіз ефективності вейвлет-перетворень при обробці біомедичних зображень. Оптико-електронні інформаційно-енергетичні технології. 2020. Т. 39, № 1. С. 25–32. URL: https://oeipt.vntu.edu.ua/index.php/oeipt/article/view/1234 (дата звернення: 10.10.2025).

19. Wicker S. B., Bhargava V. K. Reed-Solomon Codes and Their Applications. New York: IEEE Press, 1999. 336 p. (дата звернення: 15.10.2025).

20. Holub V., Fridrich J. Designing steganographic distortion using directional filters. IEEE International Workshop on Information Forensics and Security (WIFS). Tenerife, Spain, 2012. P. 234–239. URL: https://doi.org/10.1109/WIFS.2012.6412655 (дата звернення: 18.10.2025).

21. Kumar S., Muttoo S. K. A robust steganography technique using DWT and ECC. Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing. Springer, 2015. P. 487–495. (дата звернення: 21.10.2025).

22. Савченко В. В. Методологія оцінювання стійкості стеганографічних систем. Кібербезпека: освіта, наука, техніка. 2019. № 2 (6). С. 45–56. URL: https://doi.org/10.28925/2663-4023.2019.6.4556 (дата звернення: 27.10.2025).

23. Cox I. J., Miller M. L., Bloom J. A., Fridrich J., Kalker T. Digital Watermarking and Steganography. 2nd ed. Morgan Kaufmann, 2007. 624 p. (дата звернення: 30.10.2025).

24. Дубчак І. В. Застосування кодів корекції помилок в сучасних системах передачі даних. Інформаційні технології та комп'ютерна інженерія. 2021. № 1 (50). С. 34–41. URL: https://itce.vntu.edu.ua/index.php/itce/article/view/1000 (дата звернення: 01.10.2025).

25. ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги (ISO/IEC 27001:2013, IDT).

Київ : ДП «УкрНДНЦ», 2016. 32 с. URL: http://online.budstandart.com/ua/catalog/doc-page?id_doc=62345 (дата звернення: 05.10.2025).

26. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» : від 05.07.1994 № 80/94-ВР. Відомості Верховної Ради України. 1994. № 31. Ст. 286. URL: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80 (дата звернення: 10.10.2025)

27. Савченко В. В. Використання кодів корекції помилок для підвищення надійності стеганографічних систем. Наукові праці ВНТУ. 2023. № 4. С. 12–18. URL: https://praci.vntu.edu.ua/index.php/praci/article/view/567 (дата звернення: 15.10.2025).

28. Westfeld A. F5—A Steganographic Algorithm: High Capacity Despite Better Steganalysis. Information Hiding. 2001. LNCS 2137. P. 289–302. URL: https://doi.org/10.1007/3-540-45496-9_21 (дата звернення: 19.10.2025).

29. Holub V., Fridrich J. Designing Steganographic Distortion Using Directional Filters. IEEE International Workshop on Information Forensics and Security. 2012. P. 234–239. URL: https://doi.org/10.1109/WIFS.2012.6412655 (дата звернення: 27.10.2025).

30. Python Software Foundation. Python 3.11.5 Documentation. URL: https://docs.python.org/3/ (дата звернення: 30.10.2025).

ДОДАТКИ

130

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

"_22_" __вересня_____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

«Підвищення стійкості методу приховування інформації у зображеннях до пасивних атак на основі високочастотних коефіцієнтів dwt та кодів корекції помилок»

08-72.МКР.011.00.000.ТЗ

Керівник магістерської кваліфікаційної роботи
к.т.н., доц., зав. каф. МБІС

Карпінець В. В.

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб підвищення стійкості приховування даних у цифрових зображеннях до пасивних атак на основі високочастотних коефіцієнтів DWT та кодів корекції помилок.

Область застосування: системи кібербезпеки, захист авторських прав (Digital Rights Management), приховані канали передавання службових даних, захист мультимедійного контенту в ненадійних каналах зв’язку..

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №96 від 20. 03. 2025 р.

3. Мета та призначення розробки

3.1 Мета розробки: створення методу та програмного засобу для надійного приховування даних у цифрових зображеннях, що забезпечує відновлення інформації після впливу пасивних атак (JPEG-стиснення, фільтрація, шум) шляхом вбудовування у високочастотні коефіцієнти дискретного вейвлет-перетворення (DWT) та застосування кодів корекції помилок (ECC).

3.2 Призначення: вбудовування конфіденційної інформації у зображення-контейнери, забезпечення цілісності та конфіденційності прихованих повідомлень, оцінка стійкості до деструктивних впливів.

4. Джерела розробки

4.1. Petitcolas F. A. P., Anderson R. J., Kuhn M. G. Information hiding-a survey. Proceedings of the IEEE. 1999. T. 87, № 7. C. 1062–1078.

4.2. Kumar S., Singh A. A robust DWT-based steganography using error correcting codes. International Journal of Computer Network and Information Security. 2020. Vol. 12, No. 5. P. 23–35.

4.3. Mallat S. A Wavelet Tour of Signal Processing: The Sparse Way. 3rd ed. Burlington : Academic Press, 2009.

4.4. Reed I. S., Solomon G. Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics. 1960. Vol. 8, No. 2.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен реалізовувати інтерфейс для завантаження зображень, введення повідомлення та налаштування параметрів вбудовування (ключ, рівень ECC);

5.1.2 Реалізація повинна забезпечувати DWT-декомпозицію зображення та адаптивний вибір коефіцієнтів;

5.1.3 Програма повинна виконувати кодування повідомлення кодами Ріда-Соломона перед вбудовуванням;

5.1.4 Програмний засіб повинен забезпечувати витягування та корекцію помилок у повідомленні зі спотвореного стегозображення;

5.1.5 Система повинна розраховувати метрики якості (PSNR, SSIM, BER).

5.2 Вимоги до надійності:

5.2.1 Програма повинна забезпечувати стабільну роботу при обробці зображень різних форматів (PNG, BMP, JPEG) та розмірів;

5.2.2 Має бути реалізована обробка виключних ситуацій (невірний формат файлу, перевищення ємності контейнера);

5.2.3 Програмний засіб повинен коректно повідомляти користувача про неможливість відновлення даних у разі критичного пошкодження контейнера.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – з тактовою частотою не менше 2.0 ГГц (Intel Core i3/i5/i7 або аналог AMD);

– оперативна пам’ять – не менше 8 Gb;

– вільне місце на диску – не менше 500 Mb;

– середовище функціонування – операційна система сімейства Windows 10/11 або Linux; наявність інтерпретатора Python 3.8+;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам.

6. Вимоги до програмної документації

6.1 Обов’язкова наявність інструкції користувача з описом процесів вбудовування, атаки та витягування даних, наведена у відповідному розділі пояснювальної записки.

7. Вимоги до технічного захисту інформації

7.1 Забезпечення конфіденційності вбудовування через використання стеганографічного ключа.

7.2 Неможливість витягування прихованої інформації сторонніми особами без знання параметрів генерації псевдовипадкової послідовності.

8. Техніко-економічні показники

8.1 Економічна ефективність від впровадження програмного засобу повинна забезпечуватися за рахунок захисту інтелектуальної власності та конфіденційних даних.

8.2 Програмний продукт повинен бути конкурентоспроможним у порівнянні з існуючими аналогами за критерієм «стійкість/непомітність».

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми		
2	Аналіз предметної області обраної теми		
3	Апробація отриманих результатів		
4	Розробка алгоритму роботи		
5	Написання магістерської роботи на основі розробленої теми		
6	Розробка економічної частини		
7	Передзахист магістерської кваліфікаційної роботи		
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи		
9	Захист магістерської кваліфікаційної роботи		

134

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 _____Костюк І.А.

Додаток Б Лістинг програми

```
"""
utils.py
Утилітарні функції для стеганографічної системи
"""

import numpy as np
import cv2
from PIL import Image
from skimage.metrics import structural_similarity as ssim
import random

def load_image(image_path, grayscale=True):
    if grayscale:
        img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    else:
        img = cv2.imread(image_path)
    if img is None:
        raise ValueError(f"Не вдалося завантажити зображення: {image_path}")
    return img

def save_image(image, output_path):
    cv2.imwrite(output_path, image)

def calculate_psnr(original, modified):
    mse = np.mean((original.astype(float) - modified.astype(float)) ** 2)
    if mse == 0:
        return 100.0
    max_pixel = 255.0
    psnr = 10 * np.log10((max_pixel ** 2) / mse)
    return psnr

def calculate_ssim(original, modified):
    return ssim(original, modified, data_range=255)

def calculate_ber(original_bits, extracted_bits):
    if len(original_bits) != len(extracted_bits):
        raise ValueError("Довжини послідовностей не співпадають!")
    errors = sum(o != e for o, e in zip(original_bits, extracted_bits))
    ber = (errors / len(original_bits)) * 100.0
    return ber

def bytes_to_bits(byte_data):
    bits = []
    for byte in byte_data:
        for i in range(7, -1, -1):
            bits.append((byte >> i) & 1)
    return bits

def bits_to_bytes(bit_list):
    while len(bit_list) % 8 != 0:
        bit_list.append(0)
    byte_array = []
    for i in range(0, len(bit_list), 8):
        byte = 0
        for j in range(8):
            byte = (byte << 1) | bit_list[i + j]
```

```
        byte_array.append(byte)
    return bytes(byte_array)

def add_header(message_bits, message_length):
    length_bits = []
    for i in range(31, -1, -1):
        length_bits.append((message_length >> i) & 1)
    return length_bits + message_bits

def read_header(bits_with_header):
    length = 0
    for i in range(32):
        length = (length << 1) | bits_with_header[i]
    message_bits = bits_with_header[32 : 32 + length]
    return length, message_bits

"""
attacks.py
Модуль симуляції пасивних атак на стегозображення
"""

import numpy as np
import cv2
from PIL import Image
import io

def apply_jpeg_compression(image, quality):
    pil_image = Image.fromarray(image)
    buffer = io.BytesIO()
    pil_image.save(buffer, format='JPEG', quality=quality)
    buffer.seek(0)
    compressed_pil = Image.open(buffer)
    compressed_image = np.array(compressed_pil)
    return compressed_image

def apply_median_filter(image, kernel_size=3):
    filtered = cv2.medianBlur(image, kernel_size)
    return filtered

def apply_gaussian_filter(image, sigma=1.0):
    kernel_size = int(2 * np.ceil(3 * sigma) + 1)
    filtered = cv2.GaussianBlur(image, (kernel_size, kernel_size), sigma)
    return filtered

def apply_awgn_noise(image, sigma=5.0):
    noise = np.random.normal(0, sigma, image.shape)
    noisy = image.astype(float) + noise
    noisy = np.clip(noisy, 0, 255).astype(np.uint8)
    return noisy

"""
embedding.py
Модуль вбудовування інформації у зображення (DWT + RS)
"""

import numpy as np
import pywt
from reedsolo import RSCodec
from src.utils import bytes_to_bits, add_header, bits_to_bytes
```

```
import random

def select_positions_combined_all(num_positions, key):
    total_per_band = 128 * 128
    total_size = total_per_band * 3

    if num_positions > total_size:
        raise ValueError(f"Недостатньо позицій! Потрібно {num_positions}, доступно {total_size}")

    random.seed(key)
    indices = random.sample(range(total_size), num_positions)

    positions = []
    for idx in indices:
        if idx < total_per_band:
            positions.append(('LL2', idx // 128, idx % 128))
        elif idx < 2 * total_per_band:
            idx_local = idx - total_per_band
            positions.append(('LH2', idx_local // 128, idx_local % 128))
        else:
            idx_local = idx - 2 * total_per_band
            positions.append(('HL2', idx_local // 128, idx_local % 128))
    return positions

def quantize_coefficient(coeff, bit, delta=8.0):
    if coeff == 0:
        return delta if bit == 1 else 0

    sign = np.sign(coeff)
    abs_coeff = abs(coeff)
    level = int(np.round(abs_coeff / delta))

    if level % 2 == bit:
        modified_abs = level * delta
    else:
        modified_abs = (level + 1) * delta

    modified = sign * modified_abs
    return modified

def embed_message(cover_image, message, key, use_ecc=True, ecc_symbols=32, delta=8.0):
    message_bits = bytes_to_bits(message)
    message_with_header = add_header(message_bits, len(message_bits))

    if use_ecc:
        rs = RSCodec(ecc_symbols)
        message_bytes = bits_to_bytes(message_with_header)
        encoded_bytes = rs.encode(message_bytes)
        encoded_bits = bytes_to_bits(encoded_bytes)
    else:
        encoded_bits = message_with_header

    cover = cover_image.astype(float)
    coeffs = pywt.wavedec2(cover, wavelet='haar', level=2)
    LL2, (LH2, HL2, HH2), (LH1, HL1, HH1) = coeffs[0], coeffs[1], coeffs[2]

    positions = select_positions_combined_all(len(encoded_bits), key)

    LL2_mod, LH2_mod, HL2_mod = LL2.copy(), LH2.copy(), HL2.copy()
```

```
    for k, (subband, i, j) in enumerate(positions):
        bit = encoded_bits[k]
        if subband == 'LL2':
            LL2_mod[i, j] = quantize_coefficient(LL2[i, j], bit, delta=delta)
        elif subband == 'LH2':
            LH2_mod[i, j] = quantize_coefficient(LH2[i, j], bit, delta=delta)
        else:
            HL2_mod[i, j] = quantize_coefficient(HL2[i, j], bit, delta=delta)

    coeffs_modified = [LL2_mod, (LH2_mod, HL2_mod, HH2), (LH1, HL1, HH1)]
    stego_float = pywt.waverec2(coeffs_modified, wavelet='haar')
    stego_image = np.clip(stego_float[:cover_image.shape[0], :cover_image.shape[1]], 0, 255).astype(np.uint8)

    return stego_image

"""
extraction.py
Модуль витягування прихованої інформації
"""

import numpy as np
import pywt
from reedsolo import RSCodec, ReedSolomonError
from src.utils import bits_to_bytes, read_header, bytes_to_bits
import random

# select_positions_combined_all дублюється з embedding.py для синхронізації
def select_positions_combined_all(num_positions, key):
    # (Код функції ідентичний embedding.py)
    total_per_band = 128 * 128
    total_size = total_per_band * 3
    random.seed(key)
    indices = random.sample(range(total_size), num_positions)
    positions = []
    for idx in indices:
        if idx < total_per_band:
            positions.append(('LL2', idx // 128, idx % 128))
        elif idx < 2 * total_per_band:
            idx_local = idx - total_per_band
            positions.append(('LH2', idx_local // 128, idx_local % 128))
        else:
            idx_local = idx - 2 * total_per_band
            positions.append(('HL2', idx_local // 128, idx_local % 128))
    return positions

def dequantize_coefficient(coeff, delta=8.0):
    abs_coeff = abs(coeff)
    level = int(np.round(abs_coeff / delta))
    return level % 2

def extract_message(stego_image, key, message_length, use_ecc=True, ecc_symbols=32, delta=8.0):
    coeffs = pywt.wavedec2(stego_image.astype(float), wavelet='haar', level=2)
    LL2, (LH2, HL2, HH2) = coeffs[0], coeffs[1]

    if use_ecc:
        # Розрахунок розміру закодованого блоку
        total_info_bits = message_length + 32
        total_info_bytes = (total_info_bits + 7) // 8
```

```
        rs = RSCodec(ecc_symbols)
        encoded_bytes = len(rs.encode(b'x' * total_info_bytes))
        num_bits_to_extract = encoded_bytes * 8
    else:
        num_bits_to_extract = message_length + 32

    positions = select_positions_combined_all(num_bits_to_extract, key)
    extracted_bits = []

    for subband, i, j in positions:
        if subband == 'LL2':
            bit = dequantize_coefficient(LL2[i, j], delta=delta)
        elif subband == 'LH2':
            bit = dequantize_coefficient(LH2[i, j], delta=delta)
        else:
            bit = dequantize_coefficient(HL2[i, j], delta=delta)
        extracted_bits.append(bit)

    if use_ecc:
        try:
            rs = RSCodec(ecc_symbols)
            extracted_bytes = bits_to_bytes(extracted_bits)
            decoded_bytes = rs.decode(extracted_bytes)[0]
            decoded_bits = bytes_to_bits(decoded_bytes)
        except ReedSolomonError:
            return None
    else:
        decoded_bits = extracted_bits

    length, message_bits = read_header(decoded_bits)
    return bits_to_bytes(message_bits)

"""
main.py
Головний скрипт демонстрації системи
"""

import sys
sys.path.append('src')
import numpy as np
import cv2
from src.utils import save_image, calculate_psnr, calculate_ssim, bytes_to_bits
from src.embedding import embed_message
from src.extraction import extract_message
from src.attacks import apply_jpeg_compression

def demo_basic():
    print("ДЕМОНСТРАЦІЯ: Базова робота системи")
    try:
        cover = cv2.imread("images/cover/test_textured.png", cv2.IMREAD_GRAYSCALE)
    except:
        cover = np.random.randint(0, 256, (512, 512), dtype=np.uint8)

    secret_message = b"This is a CONFIDENTIAL message for testing!"
    stego_key = 123456789

    # Вбудовування
    print(f"Вбудовування повідомлення: {len(secret_message)} байт")
    stego = embed_message(cover, secret_message, stego_key, use_ecc=True, ecc_symbols=32, delta=8.0)
```

```
    psnr = calculate_psnr(cover, stego)
    print(f"Якість стегозображення (PSNR): {psnr:.2f} dB")

    # Атака (JPEG Q=90)
    print("Застосування атаки JPEG (Quality=90)...")
    attacked = apply_jpeg_compression(stego, 90)

    # Витягування
    print("Спроба відновлення...")
    message_length = len(bytes_to_bits(secret_message))
    recovered = extract_message(attacked, stego_key, message_length, use_ecc=True, ecc_symbols=32, delta=8.0)

    if recovered == secret_message:
        print("☑ Повідомлення успішно відновлено!")
    else:
        print("❌ Помилка відновлення.")

if __name__ == "__main__":
    demo_basic()

"""
app.py
Веб-додаток для демонстрації стеганографічної системи DWT+RS
"""

import streamlit as st
import sys
sys.path.append('src')

import numpy as np
import cv2
from PIL import Image as PILImage
import time

from src.embedding import embed_message
from src.extraction import extract_message
from src.attacks import apply_jpeg_compression
from src.utils import bytes_to_bits, calculate_psnr, calculate_ssim

# Налаштування сторінки
st.set_page_config(
    page_title="🔐 DWT+RS Steganography",
    page_icon="🔐",
    layout="wide"
)

# Стилі CSS
st.markdown("""
<style>
    .main-header {
        background: linear-gradient(135deg, #1e3c72 0%, #2a5298 50%, #7e22ce 100%);
        padding: 2.5rem;
        border-radius: 16px;
        color: white;
        text-align: center;
        margin-bottom: 2rem;
    }
```

```
    .metric-box {
        background: white;
        padding: 1.2rem;
        border-radius: 12px;
        box-shadow: 0 2px 8px rgba(0,0,0,0.08);
        text-align: center;
        border-top: 3px solid #2a5298;
    }
</style>
""", unsafe_allow_html=True)

# Заголовок
st.markdown("""
<div class="main-header">
    <h1 style="margin: 0;">🔐 Стеганографія DWT + Reed-Solomon</h1>
    <p>Підвищення стійкості до пасивних атак</p>
</div>
""", unsafe_allow_html=True)

# Сайдбар налаштувань
with st.sidebar:
    st.header("⚙️ Параметри")

    ecc = st.selectbox(
        "🛡️ Код корекції:",
        ["RS(255,223) — базовий", "RS(255,191) — агресивний"],
        help="Базовий: до 6% помилок, Агресивний: до 12%"
    )
    ecc_symbols = 32 if "базовий" in ecc else 64

    delta = st.slider("🎚️ Крок квантування Δ:", 4.0, 16.0, 8.0, 2.0)

    key = st.number_input("🔑 Ключ:", value=123456789, step=1)

    st.info("""
    **Характеристики:**
    - Вейвлет: Haar (2 рівні)
    - Піддіапазони: LL2+LH2+HL2
    - Стійкість: JPEG Q≥90
    """)

# Вкладки
tab1, tab2, tab3, tab4 = st.tabs(["📥 Вбудувати", "📤 Витягнути", "⚔️ Атаки", "📊 Графіки"])

# Вкладка 1: Вбудовування
with tab1:
    col1, col2 = st.columns(2)

    with col1:
        st.subheader("Зображення")
        up = st.file_uploader("Завантажити (PNG/JPG)", type=['png', 'jpg'])
        if up:
            file_bytes = np.asarray(bytearray(up.read()), dtype=np.uint8)
            cover = cv2.imdecode(file_bytes, cv2.IMREAD_GRAYSCALE)
            cover = cv2.resize(cover, (512, 512)) if cover.shape != (512, 512) else cover
        else:
            # Дефолтне зображення
            cover = np.random.randint(0, 256, (512, 512), dtype=np.uint8)
```

```
        st.image(cover, caption="Контейнер", width=350)

    with col2:
        st.subheader("Повідомлення")
        msg_text = st.text_area("Текст:", "This is a CONFIDENTIAL message!")
        message_bytes = msg_text.encode('utf-8')

        if st.button("🚀 ВБУДУВАТИ"):
            with st.spinner("Обробка..."):
                start = time.time()
                stego = embed_message(cover, message_bytes, key, use_ecc=True,
                                      ecc_symbols=ecc_symbols, delta=delta)
                proc_time = time.time() - start

                # Розрахунок метрик
                psnr = calculate_psnr(cover, stego)
                ssim = calculate_ssim(cover, stego)

                # Збереження в сесію
                st.session_state.update({
                    'stego': stego, 'key': key, 'ecc': ecc_symbols,
                    'delta': delta, 'message': message_bytes
                })

                st.success(f"Готово за {proc_time:.3f} с!")

                # Відображення метрик
                m1, m2, m3 = st.columns(3)
                m1.metric("PSNR", f"{psnr:.2f} dB")
                m2.metric("SSIM", f"{ssim:.4f}")
                m3.metric("Payload", f"{len(message_bytes)} B")

                st.image(stego, caption=f"Стегозображення (PSNR: {psnr:.2f} dB)", width=350)

# Вкладка 2: Витягування
with tab2:
    if 'stego' in st.session_state:
        st.image(st.session_state['stego'], width=300)

        if st.button("🟢 ВИТЯГНУТИ ПОВІДОМЛЕННЯ"):
            msg_len = len(bytes_to_bits(st.session_state['message']))
            recovered = extract_message(
                st.session_state['stego'], st.session_state['key'], msg_len,
                use_ecc=True, ecc_symbols=st.session_state['ecc'], delta=st.session_state['delta']
            )

            if recovered == st.session_state['message']:
                st.success("☑ Повідомлення успішно відновлено!")
                st.code(recovered.decode())
            else:
                st.error("❌ Помилка відновлення.")
    else:
        st.warning("Спочатку виконайте вбудовування.")

# Вкладка 3: Атаки
with tab3:
    if 'stego' in st.session_state:
```

```
        q = st.slider("Якість JPEG:", 50, 100, 95)

        if st.button("⚡ Застосувати атаку"):
            attacked = apply_jpeg_compression(st.session_state['stego'], q)
            st.session_state['attacked'] = attacked
            psnr_att = calculate_psnr(st.session_state['stego'], attacked)
            st.metric("PSNR після атаки", f"{psnr_att:.2f} dB")

            # Спроба відновлення
            msg_len = len(bytes_to_bits(st.session_state['message']))
            try:
                rec_att = extract_message(
                    attacked, st.session_state['key'], msg_len,
                    use_ecc=True, ecc_symbols=st.session_state['ecc'], delta=st.session_state['delta']
                )
                if rec_att == st.session_state['message']:
                    st.success(f"☑ Витримав атаку JPEG Q={q}")
                else:
                    st.error(f"❌ Пошкоджено при Q={q}")
            except:
                st.error("Неможливо декодувати")
"""
experiments.py
Скрипт для проведення експериментальних досліджень та побудови графіків
"""

import sys
sys.path.append('src')

import numpy as np
import cv2
import matplotlib.pyplot as plt
import pandas as pd
from src.embedding import embed_message
from src.extraction import extract_message, dequantize_coefficient, select_positions_combined_all
from src.attacks import apply_jpeg_compression
from src.utils import calculate_psnr, calculate_ssim, bytes_to_bits, add_header, calculate_ber
import pywt

def experiment_1_psnr_vs_payload():
    """Дослідження залежності якості зображення від обсягу вбудованих даних"""
    print("ЕКСПЕРИМЕНТ 1: PSNR vs Payload")

    # Генерація випадкового контейнера
    cover = np.random.randint(0, 256, (512, 512), dtype=np.uint8)
    key = 111222

    sizes_bytes = [10, 50, 100, 500, 1000, 2000, 3000]
    psnr_values = []
    payloads_bpp = []

    for size in sizes_bytes:
        message = b'X' * size
        stego = embed_message(cover, message, key, use_ecc=True, ecc_symbols=32, delta=8.0)

        psnr = calculate_psnr(cover, stego)
        bpp = (size * 8) / (512 * 512)
```

```
        psnr_values.append(psnr)
        payloads_bpp.append(bpp)
        print(f"Size: {size} B, PSNR: {psnr:.2f} dB")

    # Побудова графіка
    plt.figure(figsize=(10, 6))
    plt.plot(payloads_bpp, psnr_values, 'bo-', linewidth=2)
    plt.xlabel('Payload (bits per pixel)')
    plt.ylabel('PSNR (dB)')
    plt.title('Залежність якості (PSNR) від ємності')
    plt.grid(True)
    plt.axhline(y=40, color='r', linestyle='--', label='Поріг непомітності')
    plt.legend()
    plt.savefig('results/graphs/exp1_psnr_vs_payload.png')
    print("Графік збережено.")

def experiment_2_ber_vs_jpeg():
    """Дослідження стійкості до JPEG-стиснення (порівняння з/без ECC)"""
    print("\nЕКСПЕРИМЕНТ 2: BER vs JPEG Quality")

    cover = np.random.randint(0, 256, (512, 512), dtype=np.uint8)
    message = b'X' * 100
    key = 333444
    qualities = [98, 95, 92, 90, 85, 80, 70]

    # 1. Вбудовування БЕЗ ECC
    stego_no_ecc = embed_message(cover, message, key, use_ecc=False, delta=8.0)
    ber_results = []

    # Підготовка бітів для порівняння
    msg_bits = bytes_to_bits(message)
    msg_full = add_header(msg_bits, len(msg_bits))

    for q in qualities:
        attacked = apply_jpeg_compression(stego_no_ecc, q)

        # Ручне витягування для розрахунку "сирого" BER
        coeffs = pywt.wavedec2(attacked.astype(float), 'haar', level=2)
        LL2, (LH2, HL2, HH2) = coeffs[0], coeffs[1]
        positions = select_positions_combined_all(len(msg_full), key)

        extracted = []
        for subband, i, j in positions:
            if subband == 'LL2': val = LL2[i,j]
            elif subband == 'LH2': val = LH2[i,j]
            else: val = HL2[i,j]
            extracted.append(dequantize_coefficient(val, delta=8.0))

        ber = calculate_ber(msg_full, extracted)
        ber_results.append(ber)
        print(f"Q={q}: BER={ber:.2f}%")

    # Побудова графіка
    plt.figure(figsize=(10, 6))
    plt.plot(qualities, ber_results, 'r^-', label='Raw BER (без ECC)')
    plt.axhline(y=6, color='orange', linestyle='--', label='Межа корекції RS(255,223)')
    plt.xlabel('JPEG Quality')
    plt.ylabel('Bit Error Rate (%)')
```

```
    plt.title('Залежність BER від якості JPEG')
    plt.gca().invert_xaxis()
    plt.grid(True)
    plt.legend()
    plt.savefig('results/graphs/exp2_ber_vs_jpeg.png')
    print("Графік збережено.")

if __name__ == "__main__":
    experiment_1_psnr_vs_payload()
    experiment_2_ber_vs_jpeg()
```

Додаток В Ілюстративний матеріал

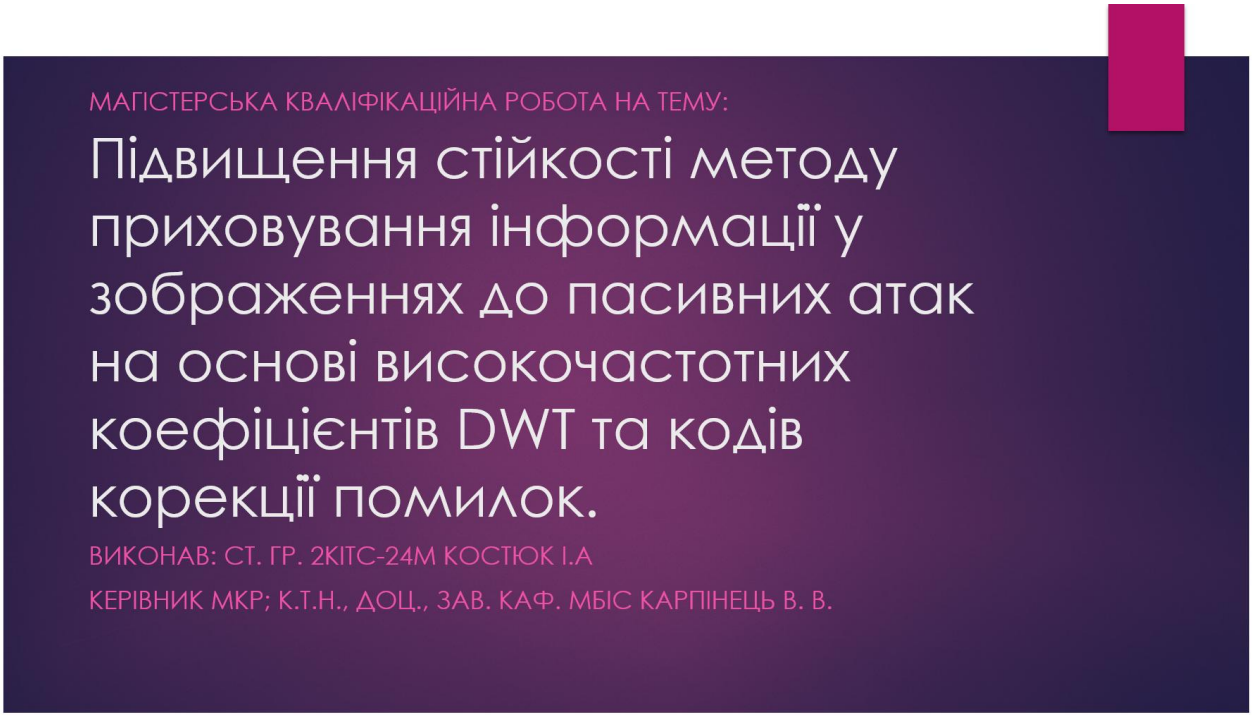

Рисунок В.1 - Слайд 1: Титульний слайд

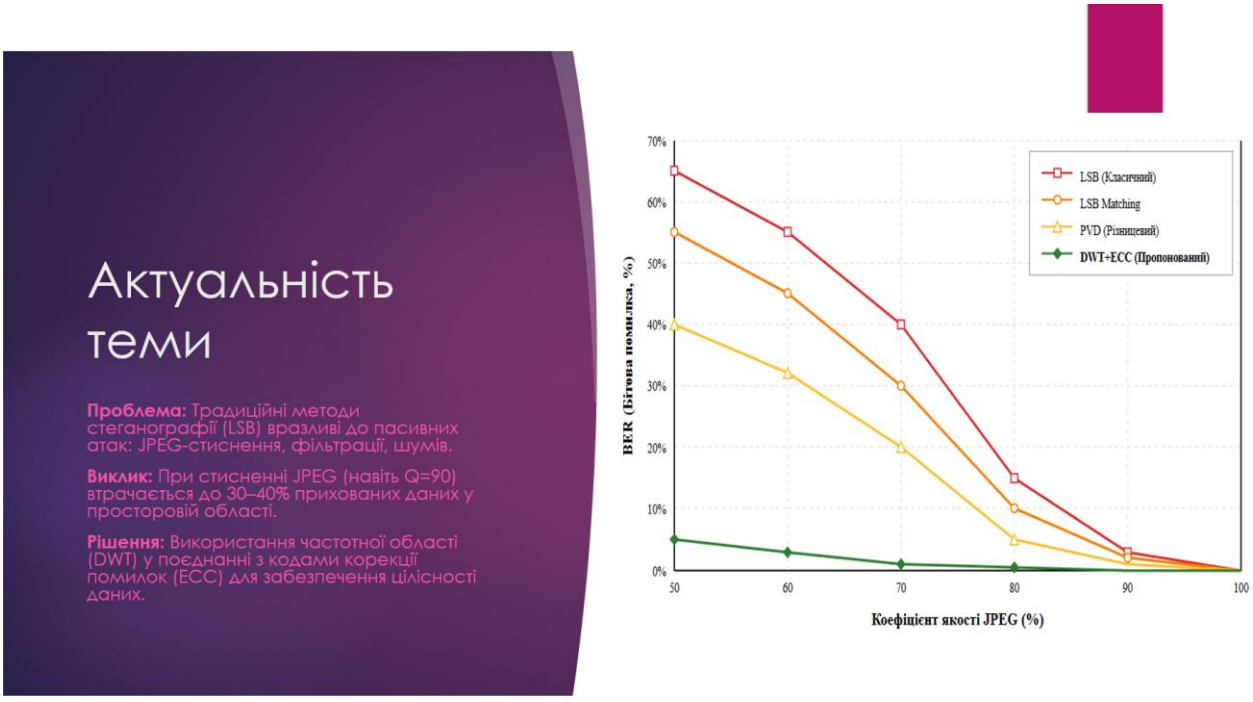

Рисунок В.2 - Слайд 2: Актуальність теми

Рисунок В.3 - Слайд 3: Мета, об’єкт та предмет дослідження

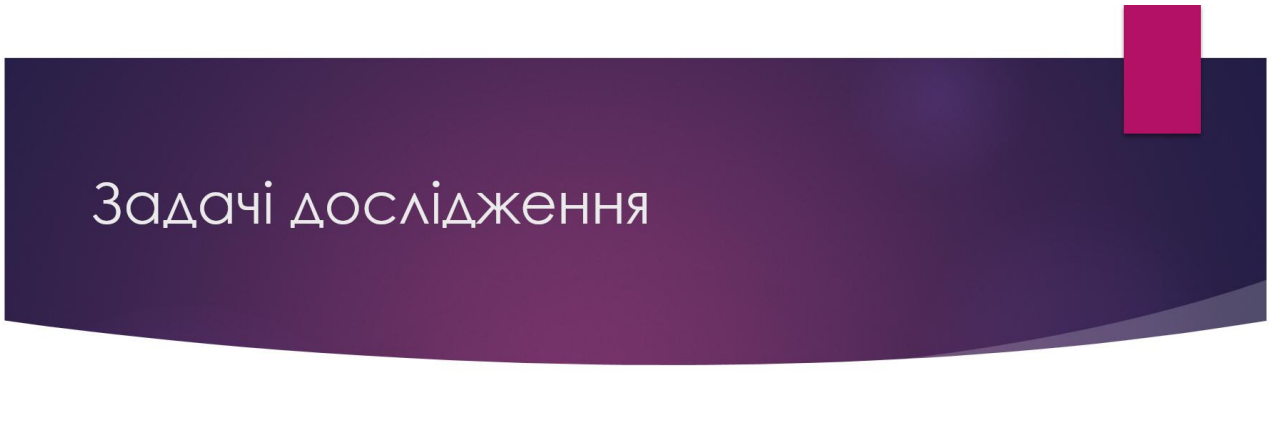

Рисунок В.4 - Слайд 4: Задачі дослідження

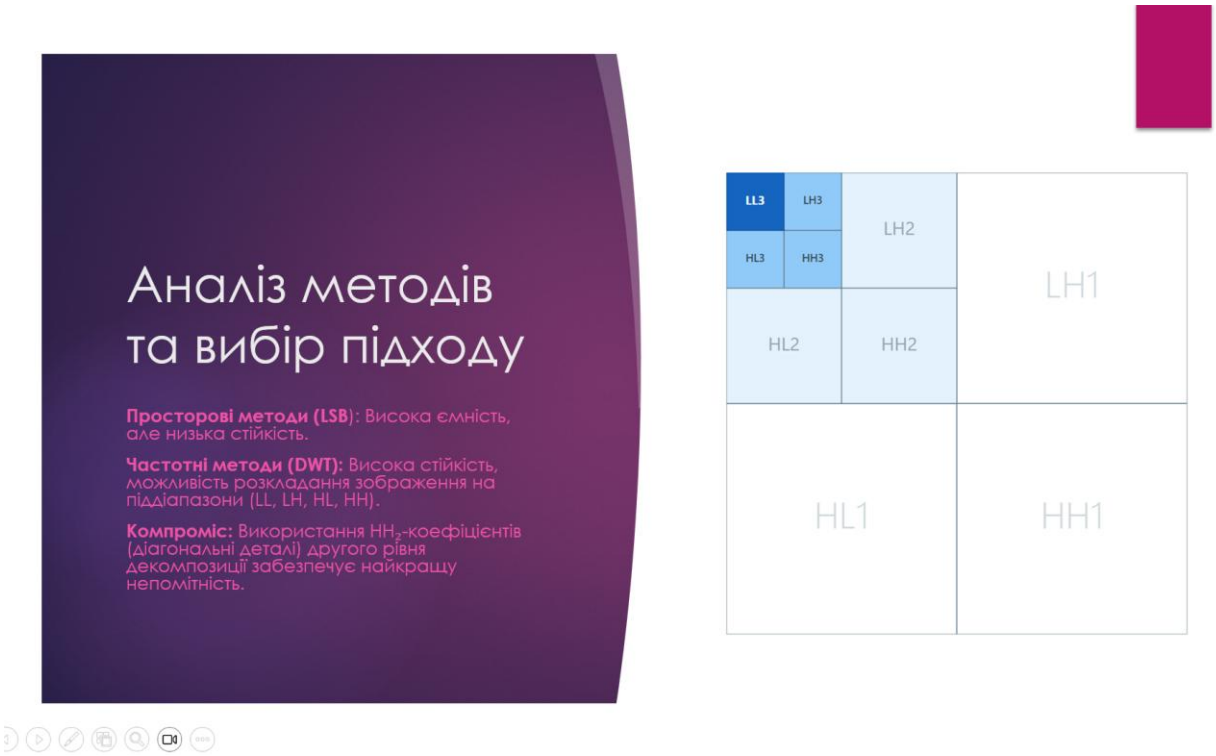

Рисунок В.5 - Слайд 5: Аналіз методів та вибір підходу

Рисунок В.6 - Слайд 6: Розроблений метод (Алгоритм вбудовування)

Рисунок В.7 - Слайд 7: Підвищення стійкості (Коди Ріда-Соломона)

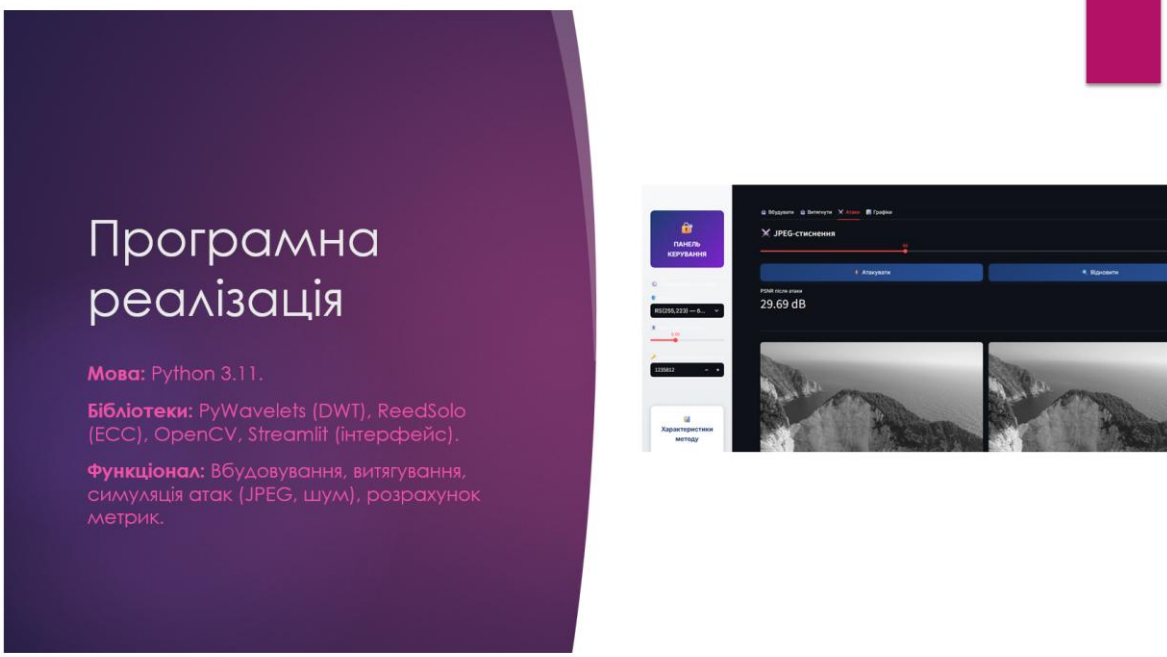

Рисунок В.8 - Слайд 8: Програмна реалізація

Результати експериментів (Непомітність)

Візуальна якість: PSNR > 42 дБ (при нормі > 40 дБ).

Структурна подібність: SSIM > 0.99 (змін не видно оком).

Ємність: до 3000 байт при збереженні високої якості.

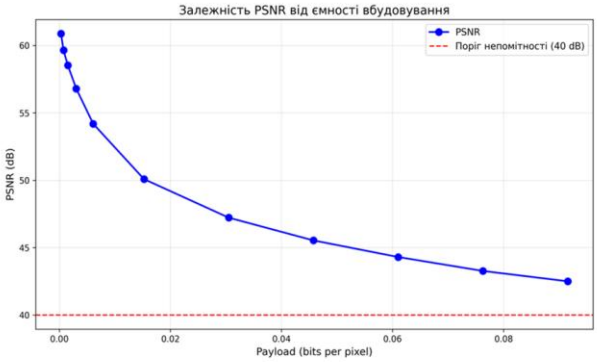

Рисунок В.9 - Слайд 9: Результати експериментів (Непомітність)

Результати експериментів (Стійкість)

Стійкість до JPEG:

1) Без ECC: BER ~12-18% (дані втрачено).

2) З ECC: повне відновлення при якості Q ≥ 70.

Висновок: Комбінація піддіапазонів LL2+LH2+HL2 показала найкращий баланс стійкості (BER 0.84% при Q=95, що легко виправляється кодом).

Піддіапазон	BER при Q=95	BER при Q=90	PSNR	Рекомендація
HH2	3.12%	20.31%	62 dB	Надто чутливий
HL2	0.00%	4.69%	70 dB	Оптимальний
LL2+LH2+HL2	0.84%	4.33%	58 dB	Обрано

Рисунок В.10 - Слайд 10: Результати експериментів (Стійкість)

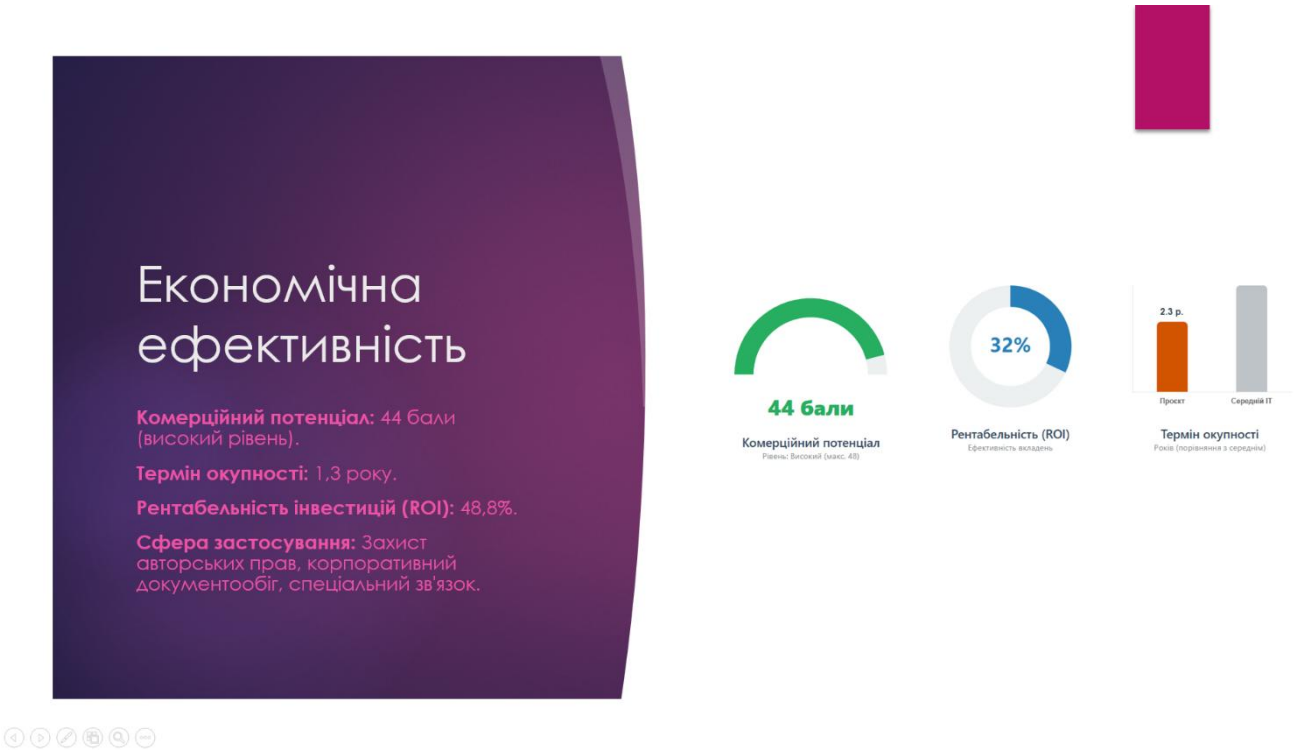

Рисунок В.11 - Слайд 11: Економічна ефективність

1) Розроблено метод на основі DWT та кодів Ріда-Соломона.
2) Досягнуто стійкості до JPEG-стиснення (Q ≥ 70) та шумів.
3) Забезпечено високу візуальну непомітність (PSNR > 42 дБ).
4) Створено програмний комплекс із веб-інтерфейсом.
5) Метод готовий до практичного використання.

Рисунок В.12 - Слайд 12: Висновки

Рисунок В.13 - Слайд 13: Дякую за увагу

Додаток Г Протокол перевірки на антиплагіат

153

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Підвищення стійкості методу приховування інформації у зображеннях до пасивних атак на основі високочастотних коефіцієнтів DWT та кодів корекції помилок

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем
факультет менеджменту та інформаційної безпеки
гр.2КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **1,17 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту**

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

к.т.н., доцент, зав. каф. МБІС Карпінець В.В. _____

к.ф.-м.н., доцент каф. МБІС Шиян А.А. _____

Особа, відповідальна за перевірку Коваль Н.П. _____

З висновком експертної комісії ознайомлений(-на)

Керівник _____ доц. Карпінець В.В.

Здобувач _____ Костюк І.А.