

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:

**«РОЗРОБКА СИСТЕМИ ЗАХИСТУ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ
ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ З ВИКОРИСТАННЯМ ЧАСТОТНИХ
ПЕРЕТВОРЕНЬ НА PYTHON»**

Виконав: студент IV курсу, групи ІІСТ-216
спеціальності 126 – Інформаційні системи
та технології

(шифр і назва спеціальності)

Олійник О. М.

(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри АІТ

Софина О. Ю.

(прізвище та ініціали)

« 12 » 06 2025 р.

Рецензент: к.т.н., доцент кафедри САІТ

Крижановський Є. М.

(прізвище та ініціали)

« 12 » 06 2025 р.

Допущено до захисту
завідувач кафедри АІТ

« 12 » 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 126 – Інформаційні системи та технології
Освітньо-професійна програма Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

завідувач кафедри АІТ
д.т.н., проф. Бісікало О. В.

« 12 » 06 2025 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Олійнику Олександру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка системи захисту зображень за допомогою цифрових водяних знаків з використанням частотних перетворень на Python»
керівник роботи Софіна Ольга Юріївна, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «20» березня 2025 року №97

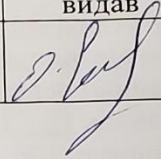
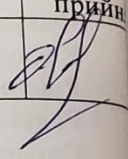
2. Термін подання студентом роботи 10.06.2025 р.

3. Вихідні дані до роботи:

- вбудовування водяного знаку за допомогою перетворення – DCT, DWT
- порівнювання оригінального та захищеного зображення через SSIM
- набір тестування захищеного зображення – обрізання, шум, стиснення
- часткове збереження зображення після атак

4. Зміст текстової частини (перелік питань, які потрібно розробити) провести огляд існуючих систем накладання водяного знаку на зображення, запропонувати структуру багатофункціонального додатку на основі частотних перетворень, розробити алгоритм роботи та програмне забезпечення для системи захисту зображень за допомогою цифрових водяних знаків
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) Система захисту зображень на основі цифрових водяних знаків, блок-схема алгоритму роботи програми; тестування різними методами використання водяних цифрових знаків; UML-діаграма послідовностей; загальна структура розробленого модуля, схема роботи програ

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Софина О.Ю., доц. каф. АПТ		

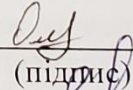
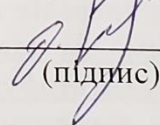
7. Дата видачі завдання 20.03.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Термін виконання		Примітка
		початок	закінчення	
1	Вибір, узгодження та затвердження теми БКР	03.10.2024	09.10.2024	Вик.
2	Аналіз предметної області та опрацювання літературних джерел.	12.03.2025	29.03.2024	Вик.
3	Постановка задач для вирішення в БКР	01.04.2025	26.04.2024	Вик.
4	Вирішення поставлених задач	29.04.2025	10.05.2024	Вик.
5	Підтвердження достовірності отриманих результатів	13.05.2025	24.05.2024	Вик.
7	Оформлення пояснювальної записки та графічної частини	27.05.2025	31.05.2025	Вик.
8	Перевірка ПЗ на відповідність вимогам	03.06.2025	07.06.2025	Вик.
9	Попередній захист, доопрацювання, та рецензування БКР	10.06.2025	15.06.2025	Вик.
10	Захист БКР ЕК	16.06.2025	16.06.2025	Вик.

Студент

Керівник роботи


(підпис)Олійник О.М.
(прізвище та ініціали)
(підпис)Софина О.Ю.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна дипломна робота складається з 89 сторінок формату А4 в тому числі і додатків, на яких є 24 рисунків, 9 таблиць, список використаних джерел містить 27 найменувань.

Присвячена розробці програмної системи для захисту цифрових зображень шляхом накладання водяних знаків. Описано основні методи маркування зображень у форматах PNG, JPEG тощо. Система дозволяє користувачеві легко та інтуїтивно вбудовувати водяні знаки з метою захисту від несанкціонованого копіювання. Програма забезпечує перевірку стійкості маркування до атак, зокрема обрізання, шуму та стиснення. Розробка виконана на мові Python із використанням бібліотеки OpenCV для обробки зображень і бібліотеки Tkinter для створення графічного інтерфейсу. Система орієнтована на масштабування та можливе комерційне застосування.

Ключові слова: водяний знак, захист зображень, Python, OpenCV, Tkinter.

ANNOTATION

The bachelor's qualification thesis consists of 89 pages of A4, including annexes with 24 drawings, 9 tables, the list of sources used contains 27 names.

It is dedicated to the development of a software system to protect digital images by applying watermarks. The basic methods of image marking in PNG, JPEG formats, etc. are described. The system allows the user to easily and intuitively install watermarks to protect against unauthorized copying. The program checks the stability of marking to attacks, including pruning, noise and compression. The development is done in Python using the OpenCV library for image processing and Tkinter Library to create a graphic interface. The system is focused on scaling and possible commercial use.

Keywords: watermark, image protection, Python, Opencv, Tkinter.

ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ І ЧАСТОТНИХ МЕТОДІВ ЗАХИСТУ ЗОБРАЖЕНЬ	6
1.1 Узагальнена класифікація стеганографічних методів	6
1.2 Розгляд основних методів цифрової стеганографії	14
1.3 Least Significant Bit - найменш значущий біт (НЗБ)	14
1.4 Дискретне косинусне перетворення	16
1.5 Метод вбудовування в область ДПФ	17
2 ОСОБЛИВОСТІ ОБ'ЄКТА ДОСЛІДЖЕННЯ	20
2.1. Метод вбудовування цифрових водяних знаків в частотну область зображення.....	20
2.2 Реалізація запропонованого методу в зображеннях формату PNG	25
2.3 Реалізація запропонованого методу в зображеннях формату JPEG	28
2.4 Аналіз запропонованого стеганографічного методу	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ЗОБРАЖЕНЬ ЗА МЕТОДОМ ЧАСТОТНИХ ПЕРЕТВОРЕНЬ РҮТНОН	34
3.1 Аргументація вибору інструменту реалізації технічного модуля	34
3.2 Розробка програмного забезпечення системи захисту зображень	42
3.3 Тестування розробленої системи захисту та аналіз його роботи.....	53
3.4 Експериментальне порівняльне тестування стійкості цифрового водяного знака	58
4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ	65
4.1. Порівняння з аналогічними системами захисту	65
4.2. Переваги та обмеження реалізованого підходу	66
4.3 Перспективи розвитку.....	68
4.4. Варіанти комерціалізації та практичного застосування	69
ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74

Додаток А (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	80
Додаток Б (обов'язковий) Лістинг основних функцій програми	84
Додаток В (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИ ЧЕНЬ	89

ВСТУП

У сучасному цифровому світі зростає значення інформаційної безпеки, особливо у сфері захисту графічного контенту. Стрімкий розвиток цифрових технологій, широке розповсюдження інтернету та мобільних пристроїв значно спростили процес створення, обробки та розповсюдження зображень. Разом із цим виникає нагальна проблема несанкціонованого використання цифрових зображень — їх копіювання, модифікації або публікації без відома чи згоди автора. Це особливо актуально в умовах зростання кількості веб-платформ, соціальних мереж, цифрових архівів і комерційних сервісів, де обіг графічного контенту є масовим.

Актуальність теми обумовлена необхідністю забезпечення захисту авторських прав та цілісності зображень, що використовуються у відкритому цифровому середовищі. З кожним роком зростає потреба у розробці надійних та ефективних інструментів, які дозволяють здійснювати маркування графічних об'єктів задля підтвердження авторства або контролю за поширенням візуального контенту.

Об'єктом дослідження є процес вбудовування та виявлення цифрових водяних знаків у зображеннях з використанням частотних перетворень. **Предметом** дослідження є алгоритми цифрового маркування на основі дискретного косинусного перетворення (DCT) та дискретного вейвлет-перетворення (DWT) та їх реалізація в Python.

Метод дослідження базується на теоретичному аналізі наявних алгоритмів цифрового маркування, що використовують частотні перетворення. Він включає експериментальне моделювання для створення та апробації програмного забезпечення, а також порівняльний аналіз результатів впровадження та видалення водяних знаків при різних видах обробки зображень: стиснення, фільтрація, кадркування та додавання шуму. Для виконання обчислювальних процесів застосовується комп'ютерне моделювання з використанням високорівневих мов програмування та спеціалізованих бібліотек

Python, таких як OpenCV, PyWavelets та NumPy. Ефективність алгоритмів оцінювалася за допомогою об'єктивних метрик подібності зображень, зокрема PSNR та SSIM.

Практичний результат роботи полягає у створенні ефективного механізму захисту зображень за допомогою цифрових водяних знаків на основі частотних перетворень. Для його досягнення необхідно виконати такі завдання:

1. Проаналізувати існуючі методи вбудовування водяних знаків у частотній області.
2. Реалізувати алгоритми вставлення та виявлення водяного знака з використанням DCT/DWT у середовищі Python.
3. Дослідити стійкість системи до різних видів обробки зображень.
4. Оцінити якість роботи алгоритму за допомогою метрик подібності (PSNR, SSIM).

Розроблена система може бути використана для захисту авторських прав на цифрові зображення від звичайних користувачів, до гігантських компаній, а також стати основою для подальших досліджень у галузі стеганографії та цифрового маркування.

Одержані результати мають як теоретичне, так і практичне значення. З теоретичної точки зору, дослідження сприяє глибшому розумінню особливостей частотного представлення зображень та принципів їх маркування.

У перспективі система може бути доповнена підтримкою інших частотних перетворень, таких як дискретне хвильове перетворення (DWT), а також удосконалена для роботи з відео та тривимірними зображеннями, що розширить її сферу застосування та підвищить рівень захисту цифрового контенту.

Апробація результатів дослідження

Початкові результати роботи було представлено на Всеукраїнській науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (2025) [10].

1 ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ І ЧАСТОТНИХ МЕТОДІВ ЗАХИСТУ ЗОБРАЖЕНЬ

1.1 Узагальнена класифікація стеганографічних методів

Останнім часом вбудовування цифрових водяних знаків (ЦВЗ) стало найбільш популярним та актуальним напрямком. Це здебільшого пов'язано з необхідністю захищати інтелектуальну власність людини від несанкціонованого розповсюдження. Створення стеганографічних каналів особливо актуальне для організацій, де безпека інформації є пріоритетом. Оскільки цілі використання цифрових водяних знаків і стеганографічних каналів відрізняються, розробники мають враховувати різні вимоги до цих методів.

Ціллю використання ЦВЗ є захист від несанкціонованого копіювання, тому знання порушником про їх наявність не є критичним. Головним викликом тут є робастність до атак, зокрема геометричних. Підвищення прихованої пропускну здатності тут не настільки важливе, як забезпечення високої достовірності прийому битів ЦВЗ [1]. Натомість у стеганографічних каналах головну роль відіграє скритність, а основними загрозами є візуальні атаки та статистичний аналіз.

Під час налаштування таких каналів важливо забезпечити достатню приховану пропускну здатність та надійність передачі інформації, незалежно від виду повідомлень. Стеганографічні методи класифікуються за типом контейнера, що підлягає модифікації: це можуть бути дані і програми, текст, фотографії та зображення. Надмірність обраного носія сприяє успішній організації прихованих вкладень, наприклад, фото дані використовуються через свою надлишковість.

Для організації прихованих каналів у ролі контейнера здебільшого використовують фото- та відеодані. Стеганографічні методи можуть бути форматними та неформатними, залежно від частини структури контейнера, яка

модифікується. Форматні методи обмежені в стеганографічній стійкості та пропускну здатності, тому більше підходять для ЦВЗ [2]. Неформатні методи, що працюють із параметрами простору приховування файлу, безпосередньо модифікують дані зображень або відео та є більш перспективними.

У даній галузі були розроблені та широко апробовані стеганографічні алгоритми, які відзначаються високою стійкістю до виявлення і забезпечують значну місткість контейнерів для прихованого розміщення повідомлень або програм. Серед них варто виділити офіційно визнані алгоритми стеганографії, зокрема DCT та DWT. Ефективність прихованої передачі інформації, що визначається рівнем прихованої пропускну здатності, безпосередньо пов'язана із надмірністю контейнера. У зв'язку з цим, найчастіше для реалізації секретного обміну інформацією використовують як статичні, так і динамічні зображення.

Процедура вибору контейнера, інтегрування прихованої інформації, і представлення цих операцій у вигляді математичних моделей детально висвітлюється в багатьох наукових роботах. Одним із базових підходів до інтеграції таких даних є принципова трансформація будь-якого контейнера у послідовність із N біт. Процес укриття інформації починається з визначення бітів контейнера, які можуть бути змінені на біти вбудовуваної послідовності без суттєвого впливу на його помітність та візуальну якість [3].

Одним із найперших і широко відомих методів впровадження прихованих даних є підхід, що ґрунтується на заміні найменш значущого біта контейнера (НЗБ). Цей метод має перевагу простоти реалізації та забезпечення максимальної пропускну здатності системи. Проте основними його недоліками залишаються низька скритність і недостатня робастність до атак або аналізу. Це стимулювало подальшу модернізацію методології НЗБ, пошук нових підходів, що супроводжувалося зростанням складності алгоритмів інтеграції та вилучення інформації разом зі зменшенням прихованої пропускну здатності.

Як зазначається в деяких дослідженнях, інтеграція даних у просторову площину цифрових зображень супроводжується появою роздвоєного піку

гістограми — важливої демаскуючої ознаки. Це визначає доцільність використання методик, орієнтованих на модифікацію спектральних компонент контейнера, які дозволяють подолати вразливість, притаманну такому типу втручань.

Стеганографічні методи, залежно від наявності ключа, поділяються на три типи: безключові, з ключем і гібридні (змішані). Для функціонування безключової стеганосистеми достатньо лише алгоритму графічного перетворення; додаткові дані, такі як стего-ключ, не потрібні. Захищеність таких систем базується виключно на секретності стеганографічних перетворень, які використовуються.

Ключова стеганосистема ділиться на дві категорії: з відкритим і закритим ключем. Системи з відкритим ключем передбачають використання закритого каналу зв'язку для передачі стего-ключа, що забезпечує вищий рівень захисту повідомлення порівняно з безключовими системами. Однак це потребує додаткових ресурсів для передачі ключа. Алгоритми з відкритим ключем працюють подібно до криптографічних систем, але головна їхня особливість полягає в тому, що ключ визначає місце вбудовування даних у контейнер, а не шифрує їх. Гібридні стеганосистеми можуть використовувати як відкритий, так і секретний ключ [4].

Залежно від способу вибору контейнера розрізняють три основні групи: сурогатні, імітаційні та селективні стеганографічні методи. У сурогатних (або безальтернативних) методах немає можливості вибору оптимального контейнера; зазвичай вибирається перший доступний контейнер, який не завжди відповідає вимогам (ерзац-контейнер). Селективні методи дозволяють відтворювати специфічні статистичні характеристики шуму прихованим повідомленням.

У цьому випадку генерується багато можливих контейнерів, а вже з них шляхом відбраковування вибирається найоптимальніший для конкретного повідомлення. Як варіант селективного підходу можна використовувати обчислення хеш-функції кожного контейнера; тоді повідомлення приховується в

тому контейнері, чия хеш-функція збігається зі значенням хеш-функції самого повідомлення.

В імітаційних методах контейнер створюється самою стеганосистемою, а його шум моделюється прихованим повідомленням. Такі системи реалізуються через спеціальні процедури, де повідомлення кодується як шум і створюється відповідна модель цього шуму. Яскравим прикладом є програма MandelSteg, яка генерує фрактал Мандельброта в ролі контейнера [5].

Контейнери за способом організації поділяються на систематичні та несистематичні. У систематичних контейнерах можливо чітко визначити місця в стеганограмі, де знаходяться інформаційні біти контейнера, а де розташовані шумові біти, призначені для маскування інформації. Так, наприклад, працює відомий і простий метод найменшого значущого біту. У разі несистематичного організованого контейнера подібне розмежування неможливе. Для витягу прихованої інформації в такому випадку необхідно аналізувати весь вміст стеганограми.

Методи доступу до прихованих даних класифікують на контейнерні способи потокового (безперервного) доступу і фіксованого (з обмеженою довжиною) доступу.

За принципом приховування даних в комп'ютерній стеганографії виділяють методи із безпосередньою заміною та спектральні методи. Перші використовують надлишковість інформації медіасередовища у просторі (наприклад, для зображень) або в часі (для звукових файлів), замінюючи малозначущі частини контейнера на біти секретного повідомлення. Натомість спектральні методи ґрунтуються на спектральному представленні компонентів середовища, куди вносяться приховані дані – як, наприклад, у коефіцієнти дискретного косинусного перетворення, перетворення Фур'є, Адамара, Хаара тощо.

Основою комп'ютерної стеганографії є використання властивостей надлишковості оригінального контейнера. Втім, важливо враховувати, що

процес приховування інформації може спричинити спотворення статистичних властивостей контейнера або порушення його структури.

Окрему категорію становлять методи, які використовують специфічні особливості файлових форматів. До них належать:

- спеціальне форматування, як-от зміщення слів, абзаців чи речень або визначення певних позицій літер;
- використання зарезервованих полів файлів, що зазвичай заповнені нулями і не враховуються програмами;
- застосування незадіяних областей магнітних чи оптичних носіїв;
- видалення заголовків-ідентифікаторів тощо.

Ці методи характеризуються низьким рівнем прихованості, малою пропускнуою здатністю і невисокою продуктивністю [6].

Формалізація деяких відомих методів, які є основою для розробки нових підходів до організації стеговкладень, представлена на таблиці 1.1.

Таблиця 1.1 – Найбільш відомі підходи до організації прихованих каналів і ЦВЗ в відеоданих та фотоданих

Класифікація методів організації прихованих каналів в відеоданих			Правило вбудовування p	Пропускна здатність прихованого каналу $C_{\text{стег}}$	Скритність U	Достовірність приховуваних даних $P_{\text{пом}}$	Число прихованих каналів
1. Напрямок цифрової стеганографії – вбудовання цифрових водяних знаків (ЦВЗ) з метою підтвердження даних							
2. Напрямок цифрової стеганографії – організація прихованого каналу з метою приховування факту передачі							
В просторовій області	Найменш значущого біта		$p^1 = I + S$	$C_{\text{стег}}^1 = \max$	$U^1 = \min$	$p_{ow} = 10^{-3}$, без стиснення	1
	К.Ю.Цветков, М.В.Коровін	К.Матсуї, К.Танака, С.Осборн, Дж.Фрідріх					

Продовження таблиці 1.1

	Псевдовипадкового інтервалу (Д.Рамкуман, Дж. Симонс, В.Волошиновський)		$p^2 = I + S I_h = 1$	$C_{\text{стер}}^2 < C_{\text{стер}}^1$	$U^2 > U^1$	$p_{ow} = 10^{-3}$, без стиснення	1
	Псевдовипадкові перестановки		$p^3 = I + S h = \varphi_i$, φ_i – ПВП	$C_{\text{стер}}^3 < C_{\text{стер}}^2$	$U^3 > U^2$	$p_{ow} = 10^{-3}$, без стиснення	1
	В.І.Коржик	С.Моллег, А.Фіцман, І.Стиран					
	Блочного приховування (В.О.Хорошко, О.Д.Азаров, К.Ю.Цветков, М.В.Коровін)		$p^4 = \left(\frac{I}{D}\right) + S$, D-число блоків	$C_{\text{стер}}^4 \approx C_{\text{стер}}^3$	$U^4 > U^3$	$p_{ow} = 10^{-3}$, без стиснення	1
В частотній області	Найменш значущого біта		$p^5 = I^{\text{ДКП}} + S$	$C_{\text{стер}}^5 = m_{ax}$	$U^5 > U^1$	$p_{ow} = 10^{-3}$, без стиснення	1
	К.Ю.Цветков, В.Г.Грибунін, М.В.Коровін	Е.Кох					
	Відносної зміни		$p^6 = \left\{ \begin{matrix} S_0, \left(\frac{I^{\text{ДКП}}}{D}\right)_1 - \left(\frac{I^{\text{ДКП}}}{D}\right)_2 > P \\ S_1, \left(\frac{I^{\text{ДКП}}}{D}\right)_1 - \left(\frac{I^{\text{ДКП}}}{D}\right)_2 \end{matrix} \right\}$ Р-поріг	$C_{\text{стер}}^6 < C_{\text{стер}}^4$	$U^6 \approx U^4$	$p_{ow} = 10^{-3}$, в процесі стиснення	1
	В.І.Коржик	Е.Кох, Дж.Жао					
	Бенгама-Меммона (Д.Бенгман, Н.Меммон, Б-Л Ео, М.Юнг)		$p^7 = \left\{ \begin{matrix} S_0, \left(\frac{I^{\text{ДКП}}}{D}\right)_1 - \left(\frac{I^{\text{ДКП}}}{D}\right)_2 > P \\ S_1, \left(\frac{I^{\text{ДКП}}}{D}\right)_1 - \left(\frac{I^{\text{ДКП}}}{D}\right)_2 \end{matrix} \right\}$	$C_{\text{стер}}^7 < C_{\text{стер}}^6$	$U^7 > U^6$	$p_{ow} = 10^{-3}$, в процесі стиснення	1

Продовження таблиці 1.1

	Фрідріх		p^8 $= (I)^{ДКП}$ $+ S (I')^{ДКП}$ $< (I)^{ДКП} + 10^2 a$ <i>I'-модифіковані дані</i> <i>a-порогова функція</i>	$C_{стег}^8$ $< C_{стег}^7$	$U^7 \approx U^6$	$p_{ow} = 10^{-3}$, в процесі стиснення	1
	Розширення спектру						
	К.Ю.Цвєтков, А.Е.Корєвих, М.В.Коровін	Дж.-Р.Сміт, В.О.Коміскі	p^{10} $= (I)^{ДКП} + S \cdot \varphi_i$ φ_i – ПВП	$C_{стег}^8$ $< C_{стег}^{10}$ $< C_{стег}^1$	$U^{10} \approx U^9$	$p_{ow} = 10^{-3}$, в процесі стиснення	1
Статистичний І.Патіс, В.О.Хорошко, В.Г.Грибунін, М.Е.Шелест			$p^9 = [(I)_1^{ДКП}]^{\frac{E}{2}} \cup [(I)_2^{ДКП}]^{\frac{E}{2}}$ Е-енергія зображення	$C_{стег}^9$ $< C_{стег}^8$	$U^9 > U^8$	$p_{ow} = 10^{-3}$, в процесі стиснення	1

Розвиток методів вбудовування інформації як у просторовій, так і в частотній областях здебільшого відбувався шляхом вдосконалення алгоритмів вбудовування та розробки методів вибору біту для модифікації таким чином, щоб отриманий розподіл був максимально близьким до випадкової величини. У дослідженні було запропоновано кілька альтернативних підходів до здійснення процесу заміни даних.

Один із способів передбачає інверсію біту: бінарне значення "1" позначає перетворення 0 на 1, тоді як "0" відповідає зміні 1 на 0. Іншим методом є вбудовування інформації шляхом вставки біту безпосередньо перед біт, що підлягає модифікації. При цьому біт секретної інформації (ЦВЗ) повинен бути протилежним за значенням до відповідного біту контейнера[7].

Використання методу видалення біту передбачає заміну пар бітів "01" або "10", які повинні відповідати різним значенням біту ЦВЗ. У цьому випадку перший біт пари видаляється для впровадження інформації.

Інший підхід використовує біт-прапор, де кожний нерухомий біт контейнера слугує бітовим індикатором ЦВЗ і вказує на інверсію попереднього біту-прапора. Подібним чином функціонує також метод граничних бітів, з тією різницею, що одному біту секретної інформації відповідає непарна кількість біт

контейнера, наступних за прапором. При цьому, якщо кількість одиничних значень у цьому наборі перевищує кількість нулів, біт ЦВЗ матиме значення "1".

Також можливе вбудовування на основі табличного підходу, де для визначення значення біту ЦВЗ використовуються методики перетворення безлічі біт у єдиний біт або табличний пошук. Таблиця може динамічно змінюватися на кожному кроці або генеруватися псевдовипадковим чином. Зважаючи на те, що табличні значення відомі і кодеру, і декодеру, їх передача є непотрібною (так звана непряма динамічна таблиця).

Інноваційні підходи також включають застосування статистичних функцій, що аналізують характеристики зображення та кореляційні зв'язки між його елементами. Ці функції можуть використовуватися для визначення оптимального шляху приховання інформації (стегошлях). У цьому випадку опціонально може бути використана псевдовипадкова послідовність.

Альтернативний підхід полягає у вбудовуванні додаткової інформації через енергетичну різницю між коефіцієнтами контейнера, що відзначається незначним впливом на загальну статистику зображення.

Однією з окремих класифікаційних груп стеганографічних методів є організація цифрових водяних знаків (ЦВЗ) та стеганографічних каналів із використанням широкосмугових сигналів (ШСС). Ці підходи також відомі як методи розширення спектра. Відмінною особливістю цього підходу є те, що модифікації піддається все зображення загалом, на відміну від методів вбудовування у частотній або просторовій областях.

1.2 Розгляд основних методів цифрової стеганографії

Для визначення найбільш придатного методу для подальшої реалізації алгоритму буде проведено порівняльний аналіз наявних підходів до цифрової стеганографії. Ці методи умовно можна класифікувати на просторові та частотні. Просторові методи передбачають безпосереднє внесення змін у зображення шляхом модифікації значень кольорових характеристик та інтенсивності окремих пікселів або їхніх груп. Натомість частотні методи функціонують у спектральній області зображення, що дозволяє реалізувати приховане впровадження даних на рівні частотних компонентів.

1.3 Least Significant Bit - найменш значущий біт (НЗБ)

Просторовий метод використовують для вбудовування секретної інформації шляхом модифікації найменш значущих бітів пікселів, зазвичай останнього одного чи двох. Такий підхід мінімально змінює інтенсивність кольору пікселя, що робить можливим запис фрагментів прихованих даних, зберігаючи при цьому зовнішній вигляд зображення майже ідентичним оригіналу [8].

Кодування зображень для веб-ресурсів здійснюється за RGB-моделлю, при якій кожен піксель формується у вигляді трьох компонентів, що відображають інтенсивність базових кольорів: червоного, зеленого та синього. Інтенсивність кожного компонента кодується одним байтом (8 біт), що відповідає значенням від 0 до 255.

Приклади кодування кольорів пікселів можуть виглядати так:

- #000000 — білий піксель;
- #FFFFFF — чорний піксель;
- #FF0000 — червоний піксель;
- #00FFFF — жовтий піксель (максимальна інтенсивність зеленого та синього каналів).

У методі НЗБ найчастіше вибирають синій канал, оскільки людське око менш сприйнятливим до змін у цьому спектрі. Водночас можна використовувати й монохромні зображення, які використовують лише один доступний канал.

Для запису повідомлення в контейнер спочатку визначається стего шлях — це набір пікселів, які буде модифіковано. Далі останній біт цих пікселів змінюється відповідно до бітів кодованого повідомлення.

Наприклад, якщо слід записати третій біт із повідомлення 0110010 у синій канал пікселя #FAFAFA, то зміна значення відбудеться наступним чином:

- Початкове значення синього каналу: FA.
- Бінарне представлення: 1111 1010.
- Запис третього біта («1») у молодший значущий біт: 1111 1011.

Результат: значення кольору змінюється на #FAFAFB.

Цей метод ідеально підходить у ситуаціях, коли вміст не буде зазнавати зовнішніх змін, оскільки його стійкість є дуже низькою (крихкий метод). Однак він дозволяє впровадити значні обсяги інформації, навіть ціле додаткове зображення всередині основного (рис. 1.1).

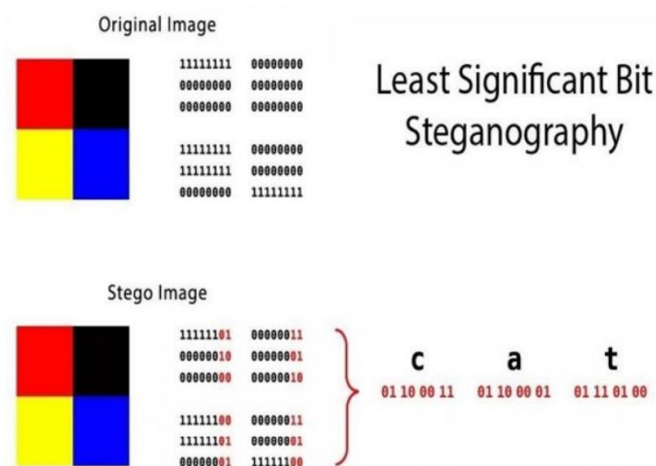


Рисунок 1.1 – Приховання тексту в зображенні 4 пікселів

Тим не менш, у випадку, коли важлива стійкість до змін, цей підхід стає непридатним, оскільки після найменших модифікацій відновлення встрайковувального цифрового водяного знака (ЦВЗ) неможливе. При цьому не

можна стверджувати, що метод узагалі непрактичний для цифрової стеганографії. Навпаки, у деяких сценаріях система потребує підтвердження автентичності контенту. Це досягається шляхом застосування водяних знаків, які при будь-якому зовнішньому втручанні руйнуються, підтверджуючи таким чином недоторканість або зміну вихідного контенту [8].

1.4 Дискретне косинусне перетворення

Метод дискретного косинусного перетворення (ДКП) дозволяє концентрувати енергію в області низьких частот, оскільки людське око є менш чутливим до високочастотних змін. Це дає можливість оцифровувати високочастотні компоненти з меншою точністю. ДКП застосовує принцип диференційного вбудовування енергії, широко використовуваного при стисненні зображень за алгоритмом JPEG та обробці відео за алгоритмом MPEG (рис 1.2).

Суть алгоритму диференційного вбудовування енергії полягає у виборі n блоків ДКП, які розділяються на дві половини. Секретний біт вбудовується шляхом модифікації різниці енергій між цими половинами через зміну високочастотних коефіцієнтів ДКП. Цей підхід демонструє відмінні результати щодо збереження візуальної схожості між оригінальним і модифікованим сигналами, але відзначається складністю реалізації та значними ресурсними затратами, що впливає на тривалість виконання. Через ці причини метод був обраний для впровадження [4].

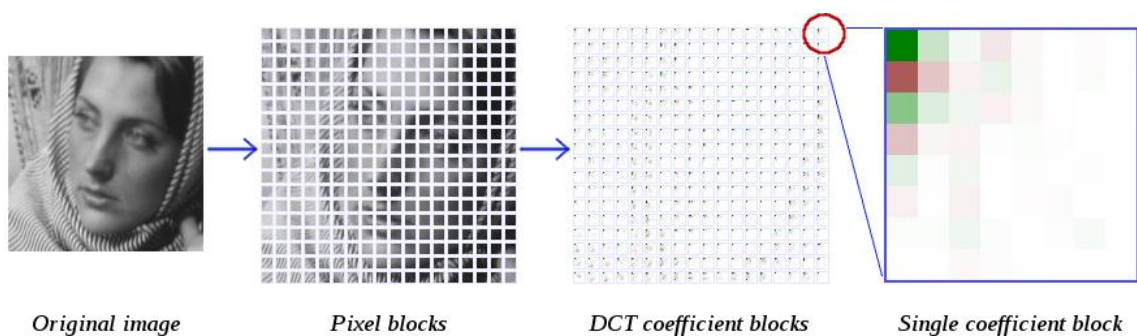


Рисунок 1.2 – Дискретне косинусне перетворення

Загалом, деякі методи характеризуються високою тривалістю виконання та складністю реалізації через забезпечення високого рівня прихованості інформації. Інші можуть не відповідати певним попередньо визначеним критеріям, як-от необхідність використання початкових зображень або недостатній обсяг даних, які можна вбудувати.

1.5 Метод вбудовування в область ДПФ

Дискретне перетворення Фур'є (ДПФ) є одним із базових понять у сфері обробки сигналів загалом, а також цифрової обробки зображень зокрема. Пряме перетворення Фур'є або Фур'є-образ $F(u)$ для неперервної функції однієї змінної $f(x)$ визначається так:

$$F(u) = \int f(x) e^{-i2\pi ux} dx \text{ (від } -\infty \text{ до } +\infty) \quad (1.1)$$

де i є уявною одиницею.

За заданим значенням перетворення Фур'є $F(u)$ вихідну функцію $f(x)$ можна відновити за допомогою зворотного перетворення Фур'є:

$$f(x) = \int F(u) e^{i2\pi ux} du \text{ (від } -\infty \text{ до } +\infty) \quad (1.2)$$

Однак у даному випадку об'єктом вивчення є дискретні функції. Для дискретної функції однієї змінної $f(x)$, де $x = 0, 1, 2, \dots, M-1$, ДПФ визначається формулою:

$$F(u) = (1/M) \sum [f(x) * e^{-(i2\pi ux)/M}] \text{ (сума по } x \text{ від } 0 \text{ до } M-1) \quad (1.3)$$

де $u = 0, 1, 2, \dots, M-1$.

Це і є пряме дискретне перетворення Фур'є. Аналогічно, вихідну дискретну функцію можна отримати за допомогою зворотного ДПФ.

Для обробки зображень, які є двовимірними масивами даних, використовується двовимірне ДПФ [10].

Це передбачає виконання ДПФ спочатку по рядках, а потім по стовпцях отриманого проміжного результату.

Перед виконанням перетворення рекомендується центрувати зображення. У результаті більша частина енергії концентрується в області низьких частот (візуально ближче до центру), що є особливо корисним при здійсненні високочастотної фільтрації зображень на (рис. 1.3).

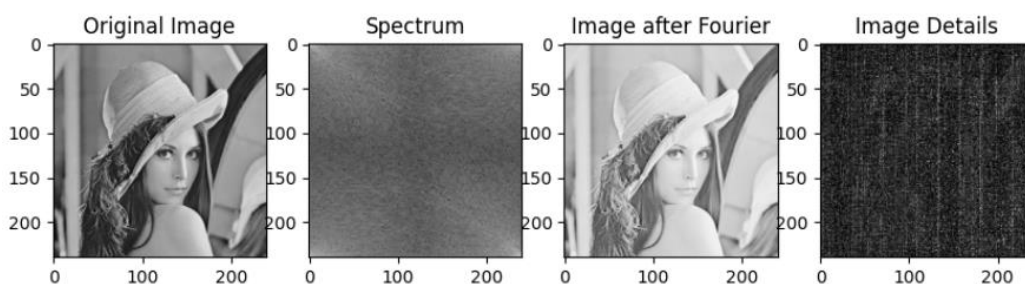


Рисунок 1.3 – Метод перетворення Фур'є та спектр логарифмічного перетворення

Запис цифрового водяного знаку (ЦВЗ) методом дискретного перетворення Фур'є (ДПФ) передбачає опрацювання зображення в частотному спектрі з метою модифікації значень таким чином, щоб під час зворотного перетворення не виникали значні видимі спотворення. Водночас, модифіковані значення повинні залишатися доступними для зчитування при повторному прямому перетворенні Фур'є у стеганографічному зображенні. Основна складність даного підходу полягає в можливості інтеграції лише напівстійких або робастних водяних знаків у частотний спектр. Спроба впровадження крихких водяних знаків може призводити до їх руйнування внаслідок обчислювальних неточностей. Зворотне перетворення із подальшим округленням згенерованих значень викликає незначні, але критичні модифікації знаку, що унеможливорює їх використання. Важливим аспектом є врахування цих обмежень на етапі розробки методики формування та запису ЦВЗ.

Однією з переваг методу є використання швидкого дискретного перетворення Фур'є (ШДПФ), високоефективного алгоритму, що забезпечує обчислення ДПФ за часову складність $O(N \log N)$. Це досягається завдяки розбиттю процесу перетворення на менші операції, що отримало назву «прорідження» [5]. Однак дана техніка накладає обмеження на розмір ряду для виконання перетворень: його довжина повинна бути кратною степеню двійки (2^k , де k – ціле число). Якщо ця умова не виконується, ряд можна доповнити нулями, при цьому необхідно врахувати цю модифікацію під час інтерпретації результату.

Робота у спектральній області зображення забезпечує гнучке управління впливом водяних знаків на результуюче зображення та дозволяє впроваджувати робастні ЦВЗ у високочастотну область спектру, де зосереджена мінімальна енергія ($<10\%$). Це сприяє забезпеченню візуальної подібності між оригіналом і модифікованим зображенням, оскільки основна інформація про зображення міститься у низькочастотному діапазоні (який концентрує максимум енергії). На основі вищезазначених факторів для подальших досліджень вибрано саме цей метод [10].

2 ОСОБЛИВОСТІ ОБ'ЄКТА ДОСЛІДЖЕННЯ

2.1. Метод вбудовування цифрових водяних знаків в частотну область зображення

Запропоновано метод, який інтегрує ЦВЗ у частотну область зображення, ґрунтуючись на застосуванні дискретного косинусного перетворення шляхом модифікації значень коефіцієнтів ДКП. За допомогою ДКП здійснюється перехід від представлення зображення як сукупності просторових хвиль до його спектральної інтерпретації. Для цього застосовуються пряме дискретне косинусне перетворення та обернене, що дозволяє переходити від частотного представлення зображення до просторового.

$$F(u, v) = \frac{c(u) \cdot c(v)}{\sqrt{2N}} \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(x, y) \cdot \cos\left(\frac{\pi \cdot u \cdot (2x+1)}{2N}\right) \cdot \cos\left(\frac{\pi \cdot v \cdot (2y+1)}{2N}\right), \quad (2.1)$$

$$S(x, y) = \frac{1}{\sqrt{2N}} \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} c(u) \cdot c(v) \cdot F(u, v) \cos\left(\frac{\pi \cdot u \cdot (2x+1)}{2N}\right) \cdot \cos\left(\frac{\pi \cdot v \cdot (2y+1)}{2N}\right), \quad (2.2)$$

де $C(x, y)$ та $S(x, y)$ – відповідно, елементи оригінального та відновленого за коефіцієнтами ДКП зображення розміром $N \times N$; x, y – просторові координати пікселів зображення; $F(u, v)$ – масив коефіцієнтів ДКП; u, v – координати в частотній області; $c(u) = \frac{1}{\sqrt{2}}$, якщо $u=0$, і $c(u)=1$, якщо $u>0$; $c(v) = \frac{1}{\sqrt{2}}$, якщо $v=0$, і $c(v)=1$, якщо $v>0$ [5].

Ефективність застосування дискретного косинус-перетворення (ДКП) у методі вбудовування цифрових водяних знаків (ЦВЗ) пояснюється тим, що ДКП моделює процес обробки зображень, враховуючи особливості зорової системи людини, виокремлюючи важливі деталі від незначущих. Це дозволяє виділити частину зображення, зміна якої забезпечить надійний захист ЦВЗ і мінімальний вплив на якість зображення. На початковому етапі первинне зображення

розділяється на блоки розміром 8×8 пікселів. Для отриманих блоків значень кожного каналу кольору в просторі RGB застосовується ДКП згідно з формулою

(1.1). В результаті виконання ДКП отримується ряд матриць коефіцієнтів базисних функцій, які містять інформацію про рівень зміни яскравості пікселів певної частоти в блоці зображення, які позначаються $Fb(u,v)$, де b – номер блоку контейнера C , а (u,v) – позиція коефіцієнта в цьому блоці. В кожній матриці низькочастотні

Коефіцієнти низької частоти (НЧ) будуть розташовані у верхньому лівому куті, високочастотні (ВЧ) коефіцієнти – у нижньому правому, а середньочастотні (СЧ) компоненти – по діагоналі матриці (рис. 2.1). НЧ коефіцієнти містять основну енергію зображення, оскільки більшість зображень складаються з низькочастотних та середньочастотних компонентів, тому основна інформація розміщується в лівій верхній частині матриць коефіцієнтів [9]. Зображення з високими значеннями ВЧ коефіцієнтів практично не зустрічаються через їх низьку інформативність.

	0	1	2	3	4	5	6	7
0	-377	-39	22	-14	7	10	-9	12
1	-12	-9	-9	33	-8	4	-5	6
2	-52	4	12	-16	-8	6	-2	2
3	-17	-10	7	11	-4	3	-3	0
4	12	4	1	3	6	-1	0	0
5	-1	2	1	5	1	2	-2	1
6	2	2	3	-2	0	1	0	0
7	-2	-2	-1	1	-1	0	0	-1

	низькочастотні коефіцієнти
	середньочастотні коефіцієнти
	високочастотні коефіцієнти

Рисунок 2.1 – Приклад коефіцієнтів матриці ДКП

Перед здійсненням приховування проводиться аналіз блоків коефіцієнтів зображення на їх придатність для вбудовування. При цьому необхідно враховувати, що блоки не повинні мати різких коливань яскравості та не повинні

бути однорідними. Блоки, які не відповідатимуть першій умові, характеризуватимуться високими значеннями низькочастотних коефіцієнтів, тоді як більшість високочастотних коефіцієнтів блоків, що не відповідатимуть другій умові, будуть дорівнювати нулю. Таким чином, граничні значення коефіцієнтів PL та PH, які задовольнятимуть першу та другу умови відповідно, будуть визначатися за формулами:

$$\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F_b(u, v)| > P_l, \quad (2.3)$$

$$\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F_b(u, v)| > P_h, \quad (2.4)$$

де PL – граничне значення для забезпечення першої умови, PH – граничне значення для забезпечення другої умови.

Також перед вбудовуванням обумовлюється граничне значення параметра P, який визначає на скільки може бути змінений коефіцієнт. Це здійснюється для запобігання суттєвої деградації зображення у випадку значної зміни коефіцієнтів ДКП. Вибір коефіцієнтів блоку зображення для приховування повідомлення відбувається таким чином. Псевдовипадково вибираються два НЧ коефіцієнти $F_b(u_2, v_2)$ та $F_b(u_3, v_3)$, причому значення позицій (u_2, v_2) та (u_3, v_3) генеруються на основі секретного ключа k. Далі відбувається визначення придатності до зміни

вибраної пари коефіцієнтів, використовуючи граничне значення P. При цьому різниця між значеннями коефіцієнтів не повинна перевищувати P, тобто:

$$F_b(u_2, v_2) - F_b(u_3, v_3) \leq p \in \{-P, P\} \quad (2.5)$$

Якщо умова (2.5) не виконується, генеруються інші позиції коефіцієнтів до повного перебору всіх значень серед діапазону позицій НЧ компонентів. Якщо блок не задовольняє умовам (2.3) та (2.4) або коефіцієнти блоку не відповідають умові (2.5) – блок відкидається і не використовується для приховування інформації. Після цього вибираються ще два коефіцієнти $F_b(u_1, v_1)$ та $F_b(u_4, v_4)$

шляхом накладання умови з граничним значенням P на сусідні коефіцієнти відносно $Fb(u_2, v_2)$ та $Fb(u_3, v_3)$. Таким чином:

$$F_b(u_1, v_1) - F_b(u_2, v_2) \leq p \in \{-P, P\}, \quad (2.6)$$

$$F_b(u_3, v_3) - F_b(u_4, v_4) \leq p \in \{-P, P\}, \quad (2.7)$$

У випадку, коли серед сусідніх коефіцієнтів знаходиться тільки один коефіцієнт, який відповідає умові (2.6) чи (2.7), блок вважається прийнятним для вбудовування тільки одного біту ЦВЗ. Блок зображення відкидається, коли не виявлено жодного коефіцієнта, який би відповідав вищенаведеній умові.

Згідно з методом, в кожен блок коефіцієнтів розміром 8×8 можна приховати 2 біти приховуваної інформації. Інформація вбудовується шляхом зміни НЧ коефіцієнтів за такою умовою: якщо потрібно вбудувати «0», то коефіцієнт $Fb(u_1, v_1)$ або $Fb(u_4, v_4)$ змінюється таким чином, щоб він був менший від середнього арифметичного значення з коефіцієнтом $Fb(u_2, v_2)$ та $Fb(u_3, v_3)$. При вбудовуванні «1» коефіцієнт повинен бути більшим від середнього арифметичного значення з цими коефіцієнтами [10]. Умови для вбудовування двобітних комбінацій m_b мають такий вигляд:

$$\left\{ \begin{array}{l} Fb(u^1, v^1) < Fb(u^1, v^1) + Fb(u^2, v^2) + Fb(u^3, v^3) / 3 \\ Fb(u^4, v^4) < Fb(u^2, v^2) + Fb(u^3, v^3) + Fb(u^4, v^4) / 3 \end{array} \right\}, \text{ при } m_b = 00$$

$$\left\{ \begin{array}{l} Fb(u^1, v^1) < Fb(u^1, v^1) + Fb(u^2, v^2) + Fb(u^3, v^3) / 3 \\ Fb(u^4, v^4) > Fb(u^2, v^2) + Fb(u^3, v^3) + Fb(u^4, v^4) / 3 \end{array} \right\}, \text{ при } m_b = 01$$

$$\left\{ \begin{array}{l} Fb(u^1, v^1) > Fb(u^1, v^1) + Fb(u^2, v^2) + Fb(u^3, v^3) / 3 \\ Fb(u^4, v^4) < Fb(u^2, v^2) + Fb(u^3, v^3) + Fb(u^4, v^4) / 3 \end{array} \right\}, \text{ при } m_b = 10$$

$$\left\{ \begin{array}{l} Fb(u^1, v^1) > Fb(u^1, v^1) + Fb(u^2, v^2) + Fb(u^3, v^3) / 3 \\ Fb(u^4, v^4) > Fb(u^2, v^2) + Fb(u^3, v^3) + Fb(u^4, v^4) / 3 \end{array} \right\}, \text{ при } m_b = 11$$
(2.8)

Після зміни коефіцієнтів проводиться обернене дискретне косинус-перетворення над матрицями коефіцієнтів згідно з формулою (2.8) – для перетворення зображення з частотного представлення в просторове. В результаті оберненого ДКП отримується набір матриць зі значеннями кольорів RGB простору. На рисунку 2.2 зображено структуру процесу вбудовування ЦВЗ:

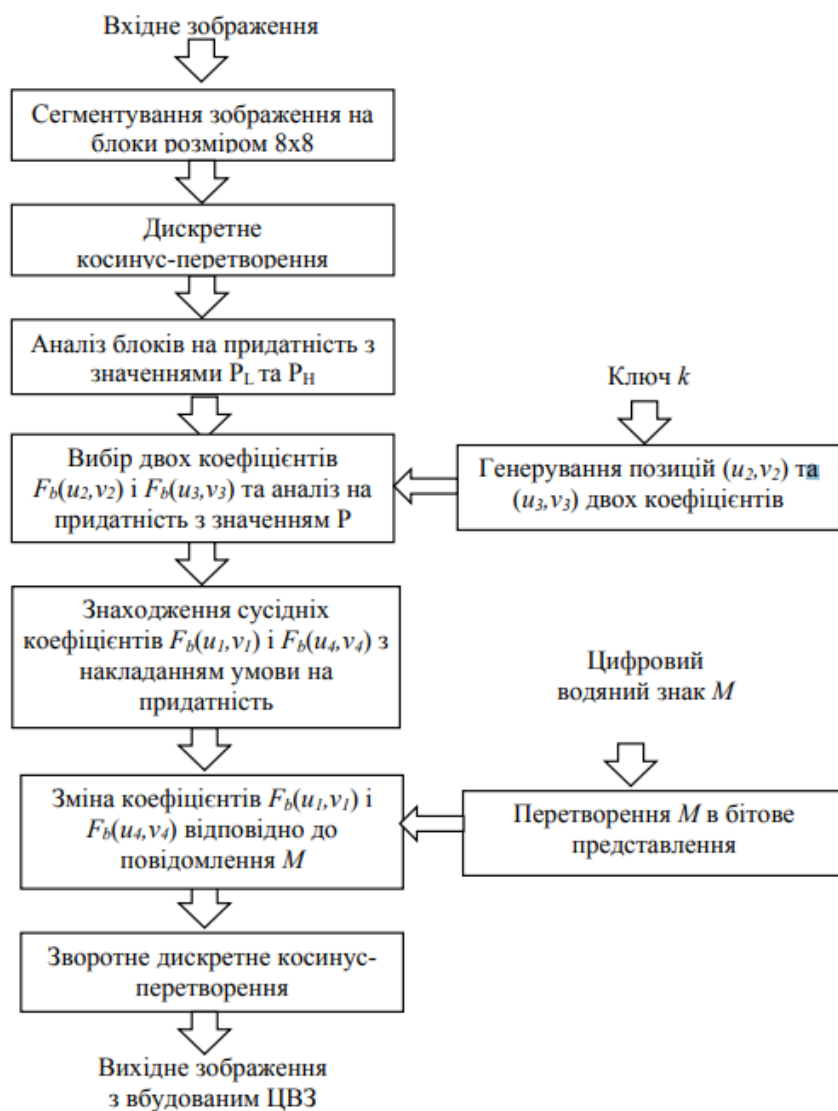


Рисунок 2.2 – Процес вбудовування цифрового водяного знаку в зображення

Виявлення цифрового водяного знаку (ЦВЗ) не потребує наявності оригінального зображення чи самого водяного знаку. Перед вилученням інформації виконується дискретне косинусне перетворення, після чого здійснюється аналіз блоків і коефіцієнтів на відповідність вимогам, визначеним

умовами (2.1)–(2.8). Для встановлення наявності певних двобітових комбінацій або окремих бітів ЦВЗ у блоці зображення здійснюється порівняння коефіцієнтів відповідно до умови (2.6).

2.2 Реалізація запропонованого методу в зображеннях формату PNG

Розроблений метод може бути ефективно адаптований і для зображень, що зберігаються у форматі PNG. Завдяки використанню безвтратного стиснення та підтримці багатьох типів кольорових моделей, PNG надає гнучке середовище для інтеграції цифрових водяних знаків (ЦВЗ). У цьому підрозділі розглянуто особливості застосування методу до зображень у форматі PNG [11].

Алгоритм стиснення у форматі PNG має такі основні етапи:

1. Попередня обробка (фільтрація): Перед основним стисненням кожен рядок зображення проходить фільтрацію. Це дозволяє зменшити обсяг повторюваних даних, покращивши результати наступних етапів. Існує кілька типів фільтрів, що застосовуються в залежності від структури даних у рядку.

2. Конкатенація та розбиття на блоки: Після фільтрації всі пікселі об'єднуються у потік байтів, який потім організовується у блоки даних згідно з внутрішньою структурою PNG-файлу. Блоки зображення зазвичай маркуються як IDAT.

3. Стиснення методом DEFLATE: Основне стиснення здійснюється за допомогою алгоритму DEFLATE, який є поєднанням кодування Лемпеля-Зіва (LZ77) та кодування Хаффмана. Цей етап дозволяє ефективно зменшити розмір зображення без втрати якості.

4. Кодування Хаффмана: На завершальному етапі DEFLATE застосовується побітове представлення символів, яке забезпечує ще компактніше зберігання даних за рахунок коротших кодів для частіше вживаних бітових шаблонів.

Для реалізації методу вбудовування цифрового водяного знаку у PNG-зображення, доцільно втручатись на рівні байтового потоку перед етапом

стиснення DEFLATE. Це дозволяє змінювати конкретні значення у піксельних даних або результатах фільтрації, зберігаючи при цьому загальну цілісність формату.

Щоб реалізувати цю інтеграцію, необхідно:

1. Розпакувати стиснені блоки IDAT: Відновити байтовий потік зображення за допомогою зворотного алгоритму DEFLATE.
2. Застосувати зворотну фільтрацію: Відновити початкові значення пікселів для точного визначення місць, придатних для вставки водяного знаку.
3. Модифікувати значення пікселів або фільтровані дані: На цьому етапі відбувається власне вбудовування водяного знаку — зміна окремих байтів відповідно до обраної схеми. Можливе використання методів на основі зміни незначущих бітів (LSB) або іншого принципу стеганографії.

Після завершення модифікації виконується повторне стиснення:

- Перефільтрація піксельних рядків;
- Стиснення методом DEFLATE;
- Формування нових блоків IDAT.

Завдяки безвтратній природі PNG-формату, дані зображення можна точно відновити після декодування, що дозволяє застосовувати методи верифікації вбудованого ЦВЗ.

Важливо підкреслити, що на відміну від JPEG, PNG не використовує перетворення у частотну область, тому метод вбудовування слід адаптувати до роботи у просторовому домені, що, втім, дає перевагу в точності й контрольованості процесу. Такий підхід дозволяє впроваджувати ЦВЗ навіть у невеликі фрагменти зображення без значущого впливу на його візуальну якість.

З міркувань стійкості до можливих спотворень або спроб видалення знаку, доцільно застосовувати додаткові заходи, зокрема дублювання певних елементів водяного знаку в різних частинах зображення або включення вбудованих механізмів перевірки цілісності. Це дозволяє підвищити надійність водяного знаку без шкоди для зображення.

Далі показано процес нанесення цифрового водяного знаку (ЦВЗ) для зображень які використовують PNG формат у вигляді схеми послідовностей, на якій поетапно чітко видно всі процеси та етапи через які проходить зображення формату PNG для того, щоб використати цифровий водяний знак на такому типі зображень як PNG (рис.2.3).



Рисунок 2.3 – Структура процесу вбудовування ЦВЗ в PNG-зображення

2.3 Реалізація запропонованого методу в зображеннях формату JPEG

Запропонований метод придатний не лише для зображень у кольоровому просторі RGB. Завдяки використанню дискретного косинусного перетворення (ДКП), його можна ефективно застосовувати для вбудовування цифрових водяних знаків (ЦВЗ) у зображення, які стискаються з використанням ДКП, зокрема за алгоритмом JPEG. Розглянемо приклад використання цього методу для зображень у форматі JPEG.

Стиснення зображень за алгоритмом JPEG включає такі етапи:

1. Перетворення кольорового простору: Зображення з простору RGB переводиться у YCbCr, де Y — яскравість, а Cb і Cr — компоненти кольоровості. Це дозволяє розділити інформацію про яскравість і кольорові відтінки пікселів.

2. Дискретизація: З урахуванням особливостей сприйняття людського ока (вища чутливість до яскравості, ніж до кольору), яскравість (Y) зберігається для кожного пікселя, тоді як Cb та Cr усереднюються для блоків 2x2 пікселів.

3. Зсув рівня: Значення компонентів Y, Cb і Cr, які спочатку є 8-бітовими беззнаковими величинами, перетворюються у знакові, шляхом віднімання 128.

4. Дискретне косинусне перетворення (ДКП): Зображення переходить зі spatial-домена (просторового представлення) у frequency-домен (частотне представлення) для подальшої обробки.

5. Квантування: Найбільш значуща стадія стиснення. Значення коефіцієнтів ДКП діляться на відповідні значення таблиці квантування, а результат округлюється. Це призводить до появи багатьох нульових коефіцієнтів, особливо у високочастотному діапазоні.

6. Кодування послідовностей нулів (RLC): Використовується для подальшого стиснення. Довгі послідовності нулів замінюються парою чисел — кількістю нулів та наступним ненульовим значенням (наприклад, послідовність 0 0 0 0 0 0 0 15 кодується як (7, 15)).

7. Кодування Хаффмана: Забезпечує компактне подання чисел у вигляді коротших бітових послідовностей, ефективніше, ніж їх стандартне двійкове представлення [12].

Запровадження методу вбудовування ЦВЗ у JPEG-зображення можливе з етапу виконання ДКП, коли алгоритм JPEG вже сформував частотне представлення у вигляді матриць ДКП-коефіцієнтів.

Для отримання цих частотних представлень з уже стиснутого JPEG-зображення необхідно здійснити його часткову декомпресію відповідно до алгоритму JPEG, виконуючи лише ті етапи, які дозволяють відновити результат ДКП без повного відтворення зображення.

1. Декодування за Хаффманом: Спочатку здійснюється зворотне перетворення — двійкові послідовності та коди Хаффмана переводяться у десяткові значення на основі відповідної таблиці Хаффмана.

2. Розкодування нульових послідовностей: Для кожної пари чисел перше значення перетворюється на послідовність нулів, що передує другому числу.

3. Зворотне квантування: Отриманий вектор чисел формується у матрицю, яка множиться на таблицю квантування, що міститься у JPEG-файлі. У результаті формується набір матриць з відновленими коефіцієнтами ДКП.

На цьому етапі здійснюється вбудовування цифрового водяного знаку шляхом модифікації коефіцієнтів ДКП відповідно до запропонованого методу.

Щоб завершити процес вставки водяного знаку та зберегти зображення у форматі JPEG, необхідно виконати етапи 5–7 алгоритму JPEG, а саме:

- провести повторне квантування;
- виконати кодування нулів (RLC);
- застосувати кодування Хаффмана.

Схема вбудовування цифрового водяного знаку у JPEG-зображення представлена на (рис 2.4).

Як видно з описаного алгоритму, для інтеграції цифрового водяного знаку у JPEG-зображення немає потреби виконувати повну процедуру зворотного перетворення JPEG.

Достатньо лише часткового розпакування до моменту отримання матриць ДКП-коефіцієнтів. Це значно прискорює процес вставки, оскільки дозволяє уникнути виконання обчислювально складних операцій прямого та зворотного ДКП, які є найбільш ресурсоемними в алгоритмі JPEG.

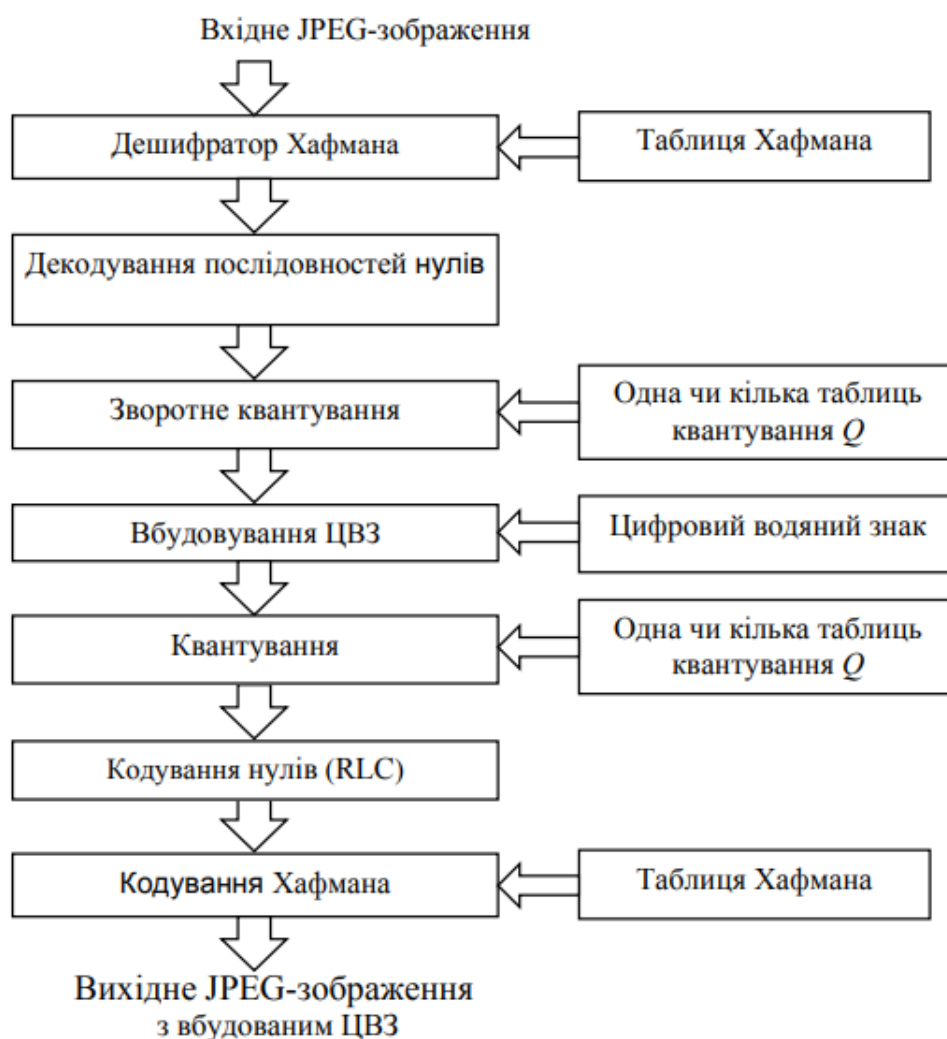


Рисунок 2.4 – Структура процесу вбудовування ЦВЗ в JPEG-зображення

2.4 Аналіз запропонованого стеганографічного методу

Оцінювання якості запропонованого стеганографічного методу здійснюється з урахуванням спотворень, які виникають у зображенні внаслідок стеганоперетворення. Для цього використовуються відповідні критерії якості [13]. Переважна більшість критеріїв, що застосовуються у візуальній обробці,

належать до групи показників відмінності, які базуються на різниці між оригінальним зображенням-контейнером і зображенням після вбудовування інформації. Інша група — це кореляційні показники, які ґрунтуються на ступені подібності між оригіналом і зображенням із вбудованим цифровим водяним знаком (ЦВЗ).

Для прикладу було обрано зображення — фотознімок природи розміром 600 Кбайт, що містить як структуровані, так і монотонні області з помірною кольоровою насиченістю. Для вбудовування використовувався графічний цифровий водяний знак обсягом 3 Кбайт. У таблиці 2.1 наведено результати оцінювання якості згідно з показниками відмінності та кореляційними критеріями для запропонованого методу, а також для відомих методів [6], які вбудовують ЦВЗ у частотну область зображення — зокрема методів Коха-Жао, Бенгама-Мемона-Ео-Юнг, Хсу-Всу та Фрідріха. У цьому випадку позначення $C_{x,y}$ відповідає пікселю вихідного зображення з координатами (x, y) , а $S_{x,y}$ — пікселю зображення з вбудованим ЦВЗ [14].

Таблиця 2.1 – Показники візуального створення при вбудуванні ЦВЗ в частотну область зображення

Назва показника	Методи вбудовування ЦВЗ в частотну область зображення				
	Коха-Жао	Бенгама-Мемона	Хсу-Всу	Фрідріх	Даний метод
Максимальна відмінність $MD = \max_{x,y} C_{x,y} - S_{x,y} $	51	60	129	101	38
Нормована середня абсолютна відмінність $NAD = \sum_{x,y} C_{x,y} - S_{x,y} / \sum_{x,y} C_{x,y} $	0.081	0.033	0.347	0.198	0.021

Продовження таблиці 2.1

Нормована середньоквадратична помилка $\text{NMSE} = \sum_{x,y} (C_{x,y} - S_{x,y})^2 / \sum_{x,y} (C_{x,y})^2$	5.343 $\cdot 10^{-3}$	2.213 $\cdot 10^{-3}$	0.032	0.015	8.056 $\cdot 10^{-4}$
Відношення “сигнал-шум” $\text{SNR} = \sum_{x,y} (C_{x,y})^2 / \sum_{x,y} (C_{x,y} - S_{x,y})^2$	187.127	634.605	24.098	43.639	806.321
Якість зображення $\text{IF} = 1 - \sum_{x,y} (C_{x,y} - S_{x,y})^2 / \sum_{x,y} (C_{x,y})^2$	0.982531	0.998721	0.9377 17	0.966753	0.999735
Якість кореляції $\text{CQ} = \sum_{x,y} C_{x,y} \cdot S_{x,y} / \sum_{x,y} C_{x,y}$	178.367	183.029	134.97 5	165.313	194.047

Оцінка ефективності та структура розробленого стеганографічного методу дають змогу визначити його переваги у порівнянні з існуючими методами, які реалізують вбудовування цифрових водяних знаків (ЦВЗ) у частотну область зображення.

Завдяки можливості приховування двох бітів інформації у кожному блоці зображення, досягається більша ємність водяного знаку. Це дозволяє або збільшити розмір ЦВЗ, або встановити більш суворі критерії до вибору блоків, придатних для модифікації.

Застосування середнього арифметичного двох частотних коефіцієнтів як критерію для вбудовування забезпечує мінімальні зміни значень коефіцієнтів дискретного косинус-перетворення (ДКП). Це, у свою чергу, дозволяє зменшити загальні спотворення вихідного зображення.

- Використання низькочастотних компонентів при вбудовуванні забезпечує підвищену стійкість ЦВЗ до стиснення, що особливо актуально при використанні форматів на кшталт JPEG [15].

Конструкція запропонованого методу також орієнтована на ефективне функціонування у стиснутих зображеннях. На прикладі алгоритму JPEG, що є найпоширенішим стандартом стиснення з втратами, продемонстровано, що цей

метод добре інтегрується в його архітектуру. Зокрема, оскільки JPEG уже використовує ДКП, можливо без додаткових обчислень скористатися готовими матрицями коефіцієнтів для вбудовування, що значно оптимізує обчислювальний процес — замість повного виконання ДКП достатньо кількох стадій алгоритму JPEG [10].

Для досягнення вищої ефективності у сфері захисту авторських прав рекомендується використовувати графічні водяні знаки, адже навіть при мінімальних спотвореннях їхній зміст залишається візуально розпізнаваним. Це забезпечує перевагу над текстовими ЦВЗ, які можуть втратити свою інформативність навіть за незначного впливу [11].

Під час розробки системи цифрових водяних знаків було обрано саме графічний водяний знак як основний об'єкт для вбудовування. Такий вибір зумовлений тим, що графічний знак (наприклад, логотип чи спеціальне зображення) дозволяє не лише підтвердити авторство, а й зберігає свою впізнаваність навіть після атак чи обробки зображення. Це особливо важливо для практичного захисту авторських прав у цифровому середовищі.

У процесі реалізації було досліджено та впроваджено два основних методи вбудовування водяного знаку: дискретне косинусне перетворення (DCT) та дискретне вейвлет-перетворення (DWT). Обидва ці методи широко використовуються у сучасних системах цифрового водяного знакування завдяки своїй стійкості до типових атак (шум, стиснення, обрізання, розмиття) та здатності забезпечувати баланс між непомітністю і надійністю вбудованого знаку [10].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ЗОБРАЖЕНЬ ЗА МЕТОДОМ ЧАСТОТНИХ ПЕРЕТВОРЕНЬ PYTHON

3.1 Аргументація вибору інструменту реалізації технічного модуля

Для зручної та ефективної реалізації модуля захисту зображень із застосуванням частотних перетворень, слід виокремити та обрати необхідні інструменти, котрі будуть оптимальними та стабільними в функціонуванні програмного модуля. Обрані інструменти повинні гарантувати стабільність системи захисту, за допомогою яких система працюватиме точно, надійно, безперебійно, незалежно від зовнішніх факторів. Крім того, інструменти мають чинити мінімальний вплив на метод вбудовування інформації, або вбудовування цифрового водяного знака (ЦВЗ) на якість самого зображення.

Щоб усі умови були враховані для обраного інструментарію, було проаналізовано три найпоширеніші мови програмування: C++, Java, Python [12].

Мова програмування C++ надає модель пам'яті та обчислень, що практично відповідає моделі більшості комп'ютерів. До того ж, вона забезпечує потужні й гнучкі механізми для абстракції; тобто мовні конструкції, котрі дозволяють програмісту створювати та використовувати нові типи об'єктів, що відповідають концепціям програми. Отже, C++ підтримує стилі програмування, котрі базуються на доволі прямому маніпулюванні апаратними ресурсами для забезпечення високого рівня ефективності, а також стилі програмування вищого рівня, що покладаються на типи, визначені користувачем, задля створення моделі даних і обчислень, яка наближена до людського розуміння завдання, котре виконує комп'ютер.

Такі високорівневі стилі програмування часто називаються абстракцією даних, об'єктно-орієнтованим програмуванням та загальним програмуванням. Варто також зазначити, що C++ також підтримує бібліотеку OpenCV, що є основою для роботи з накладанням водяних знаків на зображення.

Java - це мова програмування та потужна платформа для обчислень, вперше представлена компанією Sun Microsystems у 1995 році. З невеликого інструменту, вона еволюціонувала в ключовий елемент сучасного цифрового світу, ставши потужною платформою для багатьох сервісів та програм. Інноваційні продукти та цифрові послуги майбутнього також будуються на Java.

Хоча більшість сучасних програм вже включають середовище Java, існує ще безліч додатків і навіть веб-сайтів, які не функціонуватимуть без встановленої настільної версії Java на комп'ютері [13].

Для створення водяних знаків на зображеннях у Java використовуються бібліотеки `javax.imageio.ImageIO` для читання зображень, та клас `BufferedImage`, що використовується для обробки зображень у Java. Останній є одним з найважливіших класів для зчитування, обробки та збереження растрових зображень.

Python - це мова програмування, яка чудово підходить як для освоєння основ програмування, так і для створення реальних проєктів. Python – це об'єктно-орієнтована мова високого рівня, що має великі можливості, розроблена Гвідо ван Россумом. Завдяки зрозумілому синтаксису коду та швидкому циклу розробки, Python є ідеальним вибором для початківців та легко засвоюється. На відміну від інших мов, Python не потребує вказівки типів даних на етапі оголошення, оскільки вона має динамічну типізацію. Це полегшує процес навчання та прискорює написання коду.

Управління пам'яттю в Python відбувається автоматично, що відрізняє її, наприклад, від C++, де пам'ять необхідно звільняти самостійно. Також варто згадати про "збирач сміття" (garbage collector), який автоматично очищає пам'ять від непотрібних об'єктів. Інтерактивне середовище Python з REPL (Read-Eval-Print Loop) дозволяє тестувати код по рядках, без потреби створення цілої програми.

Python має широку підтримку різноманітних бібліотек, від розробки веб-сайтів до розробки систем машинного навчання, з легким у використанні API.

Характеристика порівняння мов програмування з критеріями: синтаксису, типізації, швидкості, навчання, портативності та застосування наведено у таблиці 3.1.

Таблиця 3.1 – Характеристика порівняння представлених мов програмування

Характеристика	Python	Java	C++
Синтаксис	Простий	Формальний	Складний
Типізація	Динамічна	Статична	Статична
Швидкість обробки	Низька	Середня	Висока
Навчання	Легке	Середнє	Складне
Управління Пам'ятю	Автоматична	Автоматична	Ручна
Портативність	Висока	Дуже висока(JVM)	Обмежена
Застосування	Python, AI, Web	Android, Backend	Ігри, Системне ПЗ, Драйвери

Після ознайомлення з критеріями зіставлення для дипломної роботи, було вирішено обрати найдоступнішу мову програмування – Python. Це зумовлено її зрозумілим кодом та різноманіттям бібліотек, які легко інтегруються з конкретною версією Python та середовищем, де відбувається розробка ПЗ. Ключовим чинником стало забезпечення гнучкості розробки: невелика кількість коду, його легке сприйняття, розробка та застосування графічного інтерфейсу системи захисту. Важливо відзначити й автоматичне управління пам'яттю, адже цей аспект суттєво прискорює процес розробки, усуваючи необхідність зосереджуватися на тонкощах використання та обмеженнях пам'яті, споживаної програмою.

В основі системи захисту зображень, що базується на частотних перетвореннях, лежить низка інструментів, які можуть бути застосовані в процесі розробки ПЗ. Від них залежить точність нанесення ЦВЗ на зображення, стабільність роботи системи та її стійкість до таких атак, як стиснення та накладання шуму. Ці атаки часто використовують зловмисники з метою привласнення авторських прав на продукт.

Тому було розглянуто декілька найпопулярніших та найсучасніших бібліотек які підтримує мова програмування Python [14].

Бібліотека SciPy представляє собою вдосконалену альтернативу NumPy, забезпечуючи розширений набір інструментів та функцій для наукових та інженерних розрахунків. Для виконання DCT, вона задіює модуль `scipy.fftpack`, де вже наявні готові функції DCT та IDCT для обчислення прямого та оберненого дискретного косинусного перетворення. Головними перевагами цієї бібліотеки є висока швидкість обчислень та легкість інтеграції з іншими бібліотеками, що використовуються в мові програмування Python.

Бібліотека PyWavelets — це ключовий компонент для застосування власне вейвлет-перетворень у Python. Її серцевиною з-поміж видів вейвлетів є: Хаара, Добеші та Сімет. Вона забезпечує можливість використання прямого та дискретного вейвлет-перетворення зображень. Цей інструмент дає змогу зручно та гнучко контролювати рівні декомпозиції, дозволяючи розміщувати водяний знак у конкретних піддіапазонах частот, що значною мірою підвищує надійність системи захисту від нападів.

NumPy - фундамент для взаємодії з багатовимірними масивами, що визначає хід виконання чисельних обчислень, найбільш вживаних в середовищі Python. Безпосередньо застосувати алгоритми DCT та DWT вона не в змозі. Проте, передбачена інтеграція з іншими бібліотеками для конвертації зображень в числове представлення, приведення значень пікселів до норми, а також для проведення операцій з великими масивами коефіцієнтів перетворень. Саме ці функціональні можливості стали основою для бібліотек SciPy, PyWavelets та їхньої синергії.

OpenCV бібліотека здебільшого слугує для опрацювання та аналізу зображень, будучи бібліотекою так званого "комп'ютерного зору". У цій роботі OpenCV застосовується для обробки вхідних зображень (зчитування, масштабування, перетворення відтінків в сірий колір зображення) та збереження результату накладання водяного знаку на зображення. Бібліотека такого типу не містить готових інструментів для DCT та DWT перетворень, проте відмінно інтегрується з наступними бібліотеками: NumPy, SciPy та PyWavelets. У таблиці 3.2 наведено загальну характеристику бібліотек щодо зручності їх використання та наявності частотних перетворень DCT, DWT [15].

Таблиця 3.2 – Порівняльна характеристика бібліотек за зручністю використання

Бібліотека	Простота синтаксису	Застосування для DCT/DWT	Інтеграція з OpenCV	Загальна зручність
SciPy	Висока	DCT	через NumPy	++++
PyWavelets	Середня	DWT	через NumPy	++++
NumPy	Висока	(базова)	Так	+++++
OpenCV	Висока	Немає підтримки	-	++++

Ось перелік плюсів та мінусів кожної бібліотеки, що були розглянуті раніше. На основі цього можна зробити вибір кількох з них, для подальшого застосування у розробці системи захисту, виходячи з конкретних вимог та умов експлуатації.

Бібліотека NumPy - це ключовий та найважливіший інструмент для обробки багатовимірних масивів даних, що критично важливо при роботі з зображеннями, котрі представляються числовими матрицями. Завдяки своїй швидкодії та підтримці від наукової спільноти, NumPy гарантує зручну та продуктивну роботу з числовими даними. Саме ці властивості перетворили її на

основу для багатьох спеціалізованих бібліотек, зокрема OpenCV, SciPy, PyWavelets та інших.

Проте важливо відзначити, що NumPy за замовчуванням не містить реалізацій дискретного косинусного перетворення (DCT) або дискретного вейвлет-перетворення (DWT). Тому для використання частотних методів в обробці зображень необхідно використовувати NumPy у комбінації з додатковими бібліотеками, які забезпечують відповідні математичні функції.

Бібліотека OpenCV – одна з найчастіше вживаних бібліотек, котра широко використовується для завдань комп'ютерного зору та обробки зображень. Вона пропонує розгалужений набір інструментів для здійснення базових операцій, як-от завантаження зображень, їх масштабування, фільтрація, відображення та збереження. Завдяки легкому синтаксису, щільній інтеграції з масивами NumPy, підтримці багатьох графічних форматів та високій продуктивності, ця бібліотека є ефективним інструментом для попередньої обробки зображень та створення прототипів систем їх аналізу [16].

Незважаючи на багатий функціонал, OpenCV не містить власної реалізації дискретного косинусного перетворення (DCT) чи дискретного вейвлет-перетворення (DWT), що трохи обмежує її використання в сфері частотної стеганографії. З огляду на це, її переважно використовують як допоміжну бібліотеку на стадіях підготовки зображень та візуалізації результатів аналізу.

PyWavelets це спеціалізована бібліотека для роботи з вейвлет-перетвореннями в Python, що робить її надзвичайно корисною для завдань частотного аналізу зображень. Вона підтримує широкий спектр вейвлет-функцій, включаючи Haar, Daubechies, Symlets, Coiflets та інші, дозволяючи підбирати оптимальну вейвлет-базу, враховуючи специфіку задачі, зокрема при реалізації методів цифрового водяного знакування. PyWavelets забезпечує виконання як одно-, так і двовимірною дискретного вейвлет-перетворення (dwt, dwt2), а також зворотних операцій (idwt, idwt2), що дозволяє здійснити повний цикл обробки зображень.

Ключовою перевагою бібліотеки є її повна сумісність з масивами NumPy, що сприяє зручній інтеграції з іншими інструментами аналізу даних, такими як OpenCV та SciPy. Крім того, підтримка багаторівневої вейвлет-декомпозиції дозволяє збільшити ефективність алгоритмів приховування інформації, зосереджуючи їх у середньо- та низькочастотних компонентах зображення, що збільшує стійкість до різних видів обробки [17].

Водночас, слід враховувати деякі обмеження використання PyWavelets. Зокрема, бібліотека не надає вбудованих засобів для візуалізації результатів, тому для перегляду оброблених зображень потрібне використання сторонніх інструментів, наприклад, Matplotlib або OpenCV. Крім того, для ефективної роботи з вейвлетами потрібні базові знання у сфері цифрової обробки сигналів, а також ретельний контроль розмірів масивів під час декомпозиції та реконструкції. У порівнянні з дискретним косинусним перетворенням (DCT), що реалізоване в SciPy, вейвлет-декомпозиція може вимагати більше обчислювальних ресурсів, особливо при використанні кількох рівнів [18].

Бібліотека SciPy – це ключовий засіб для наукових обчислень у Python, особливо в царині обробки сигналів та зображень. Модуль `scipy.fftpack` надає підтримку дискретному косинусному перетворенню (DCT) та його зворотному варіанту (IDCT), що робить бібліотеку придатною для частотних стеганографічних технік. DCT знайшло широке застосування в JPEG та методах накладання водяних знаків через його здатність зосереджувати енергію в низькочастотних складових. SciPy характеризується доброю взаємодією з NumPy, простим інтерфейсом користувача та високою точністю обчислень. Бібліотека має активну підтримку та добре розроблену документацію.

Якщо ми розглядаємо вади цієї бібліотеки в контексті частотних перетворень, то слід зауважити, що SciPy орієнтована на математичні обчислення і не пропонує готових інструментів для обробки чи представлення зображень, у зв'язку з чим необхідне використання додаткових бібліотек (OpenCV, PIL). Реалізація дискретного косинусного перетворення (DCT) в `scipy.fftpack` досить проста і не передбачає, скажімо, блокової обробки, як це

реалізовано в JPEG, що потребує самостійної реалізації. Неточності в масштабуванні здатні провокувати артефакти та погіршення якості зображення [19].

З огляду на зіставлення сильних та слабких сторін розглянутих бібліотек, постає висновок, що найліпшим варіантом для розробки софту для систем захисту цифрових зображень є їх поєднане використання — OpenCV, NumPy і PyWavelets. Бібліотека NumPy гарантує високопродуктивну взаємодію з багатовимірними масивами та числовими операціями, що є наріжним каменем обробки зображень. OpenCV пропонує великий набір функцій для базової обробки, зчитування, масштабування, фільтрації та візуалізації графічних даних, що робить її незамінним інструментом для роботи з цифровими зображеннями. PyWavelets розширює функціонал, реалізуючи частотний аналіз за допомогою вейвлет-перетворень, що є ключовим компонентом при розробці стеганографічних алгоритмів у частотній області.

Попри обмеження кожної бібліотеки, синергія від їхнього спільного використання стає потужним засобом нівелювання слабких сторін окремих інструментів. Наприклад, відсутність у OpenCV та NumPy потужного частотного аналізу легко компенсується функціоналом PyWavelets, а недостатність візуалізаційних та обробних інструментів у PyWavelets згладжується за рахунок можливостей OpenCV. Цей інтегрований підхід утворює цілісний, гнучкий та надійний набір інструментів, що сприяє адаптації алгоритмів під конкретні потреби та забезпечує високу якість захисту цифрових зображень.

Саме такий набір бібліотек було використано у ході розробки програмного забезпечення системи захисту зображень методами частотних перетворень. Такий вибір інструментів дозволяє знехтувати проблематикою однієї бібліотеки та додати потрібний функціонал іншої.

3.2 Розробка програмного забезпечення системи захисту зображень

За основу системи захисту зображень взято PyWavelets. Ця бібліотека надає підтримку дискретного вейвлет-перетворення, відомого як DWT. DWT дає змогу розкласти зображення на ділянки з різною частотою, зокрема LL, LH, HL та HH[20].

Для реалізації всіх необхідних процесів у програмному забезпеченні було застосовано OpenCV. Ця бібліотека використовувалась для: завантаження зображень (cv2.imread), переведення зображення у відтінки сірого (cv2.IMREAD_GRAYSCALE), виконання перетворення DCT (cv2.dct), масштабування та зміни розміру зображень (cv2.resize), а також для атак на зображення, наприклад, додавання шуму, розмиття та стиснення JPEG, використовуючи метод cv2.imencode. Розглянемо блок схеми алгоритму для загальної структуризації програми.

Спочатку було імпортовано всі необхідні бібліотеки та фреймворки. Це потрібно для розробки графічного інтерфейсу, масштабування зображень перед накладанням водяних знаків, а також для імітації тих атак, що можуть влаштовувати зловмисники, щоб привласнити собі авторський продукт.

```
import os
import numpy as np
import cv2
import tkinter as tk
from tkinter import filedialog, messagebox, ttk
from PIL import Image, ImageTk
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim
import pywt
```

Далі використовується головне вікно графічного інтерфейсу за допомогою бібліотеки Thinker зображено на рисунку 3.1:

```

class WatermarkApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Система цифрових водяних знаків")
        self.root.geometry("1100x750")
        self.root.configure(bg='#f0f2f5')

        # Кольори програми
        self.colors = {
            'primary': '#2a5cff',
            'primary_light': '#e6ecff',
            'secondary': '#1a1a2e',
            'background': '#f0f2f5',
            'card': '#ffffff',
            'text': '#333333',
            'success': '#28a745',
            'warning': '#ffc107',
            'danger': '#dc3545'
        }

```

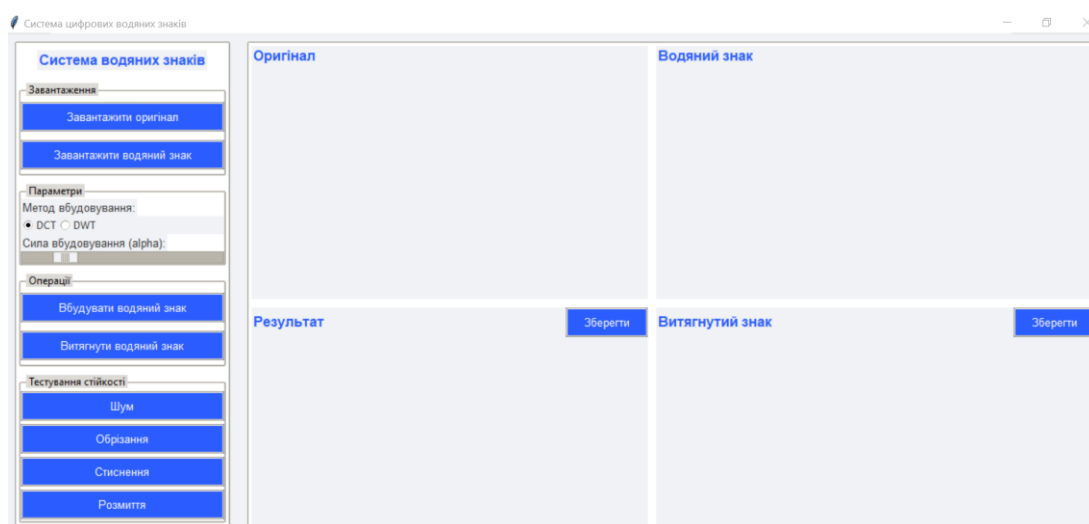


Рисунок 3.1 – Графічний інтерфейс програми використовуючи Tkinter

Клас WatermarkApp є GUI застосунком у даній програмі, root – головне вікно програми, передається ззовні, при створенні екземпляра класу. Задано

розмір головного вікна програми у 1100x750 пікселів та стилізація кольорів, фону тексту та інших компонентів. Перейдемо до основних функцій програми, це ініціалізація змінних під час роботи програми.

```
# Ініціалізація змінних
self.original_image = None
self.watermark_image = None
self.watermarked_image = None
self.extracted_watermark = None
self.alpha = 0.1
self.method = "dct"

# Налаштування стилів
self.setup_styles()

# Створення інтерфейсу
self.create_widgets()
```

У цьому шматку коду відбувається ініціалізація (`original_image`) – вихідне зображення, або як його ще називають, джерело.

Змінна (`watermark_image`) – зберігає зображення з водяним знаком, котрий було накладено на вихідне зображення. Застосування змінної (`watermarked_image`) потрібно для збереження результату нанесення водяного знаку. Змінна (`extracted_watermark`) використовується для результату видалення водяного знаку з зображення (рис.3.2).

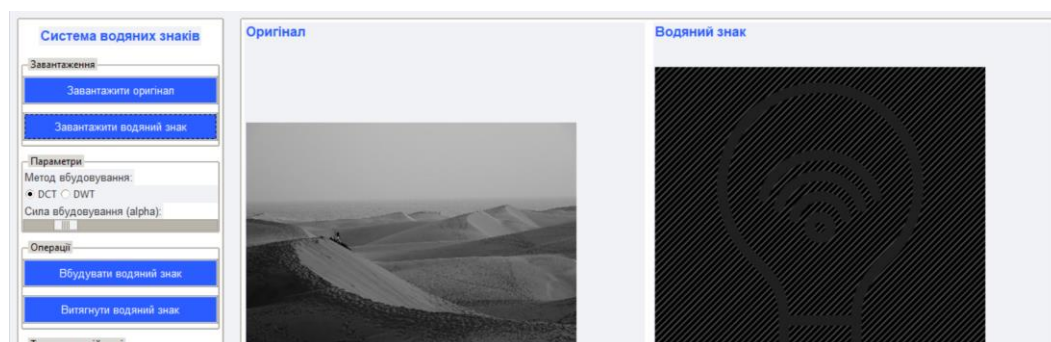


Рисунок 3.2 – Вхідна ініціалізація зображення

Також використано коефіцієнт α , що визначає прозорість водяного знаку. Чим більше значення α , тим чіткіше видно водяний знак на зображенні. Змінна `self.method = "dct"` визначає основу функціонування цієї програми, а саме перетворення зображення за допомогою дискретного косинусного перетворення DCT.

```
def update_alpha(self, val):
    self.alpha = float(val)

def load_image(self, is_watermark=False):
    filetypes = [("Зображення", "*.jpg *.jpeg *.png *.bmp *.tiff")]
    filepath = filedialog.askopenfilename(filetypes=filetypes)

    if not filepath:
        return None

    try:
        image = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
        if image is None:
            raise ValueError("Не вдалося завантажити зображення")

        if is_watermark:
            self.watermark_image = image
            self.update_display("Водяний знак", image)
        else:
            self.original_image = image
            self.update_display("Оригінал", image)

        self.status_var.set(f"Завантажено: {os.path.basename(filepath)}")
        return image
    except Exception as e:
        messagebox.showerror("Помилка", f"Не вдалося завантажити зображення: {e}")
        return None
```

Описані процедури (`update_alpha`) та (`load_image`) слугують для коригування значення прозорості (альфа) та імпортування початкового зображення чи водяного знаку у програму. Фрагмент коду `filepath = filedialog.askopenfilename(filetypes=...)`, був застосований для забезпечення підтримки форматів зображень: `jpg, jpeg, png, bmp, tiff`.

Змінна `image = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)` забезпечує завантаження картинки у відтінках сірого кольору, у випадку невдалого зчитування зображення, буде виведено помилку, що передбачено в коді. Оновлення вікна перегляду відбувається за допомогою фрагменту `self.update_display(...)`. У разі виникнення помилки, з'являється вікно з інформацією про проблему. Детальна робота з аргументами представлена у таблиці 3.3.

Таблиця 3.3 – Робота аргументу та запобігання помилки

Аргумент (<code>is_watermark</code>)	Дія	Зберігає до	Відображення у GUI
False	Завантаження оригінального зображення	<code>self.original_image</code>	У мітку «Оригінал»
True	Завантаження водяного знаку	<code>self.watermark_image</code>	У мітку «Водяний знак»

Масштабування зображення для відображення

```
h, w = image.shape
```

```
max_size = 400
```

```
scale = min(max_size / h, max_size / w)
```

```
img_resized = cv2.resize(image, (int(w * scale), int(h * scale)))
```

```
img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_GRAY2RGB)
```

```

# Конвертація для tkinter
img_pil = Image.fromarray(img_rgb)
img_tk = ImageTk.PhotoImage(img_pil)

# Оновлення мітки
label = self.labels.get(position)
if label:
    label.configure(image=img_tk)
    label.image = img_tk

```

Цей код реалізує функцію зміни розмірів зображень та їх відображення в графічному інтерфейсі користувача (GUI). Він інтегрований у метод `update_display(self, position, image)`. Основна задача цього коду – візуалізація оригінального водяного знаку, вихідного зображення та зображення з видаленим водяним знаком у відповідних віджетах програми. Нижче представлено код, який відповідає за головний метод вбудовування водяного знаку – `embed_watermark(self)`. Він спочатку перевіряє, чи завантажено оригінальне зображення та водяний знак, який планується накласти на зображення (рис.3.3).

```

def embed_watermark(self):
    if self.original_image is None or self.watermark_image is None:
        messagebox.showwarning("Увага", "Будь ласка, завантажте оригінальне зображення та водяний знак")
        return

    try:
        method = self.method_var.get()
        if method == "dct":
            self.dct_embed()
        else:
            self.dwt_embed()

        # Оновлення відображення
        self.update_display("Результат", self.watermarked_image)

```

```

# Розрахунок метрик
psnr_val,      ssim_val      =      self.calculate_metrics(self.original_image,
self.watermarked_image)

self.status_var.set(f'Вбудовано. PSNR: {psnr_val:.2f} dB, SSIM: {ssim_val:.4f}')

except Exception as e:
    messagebox.showerror("Помилка", f'Помилка при вбудовуванні: {e}')

```

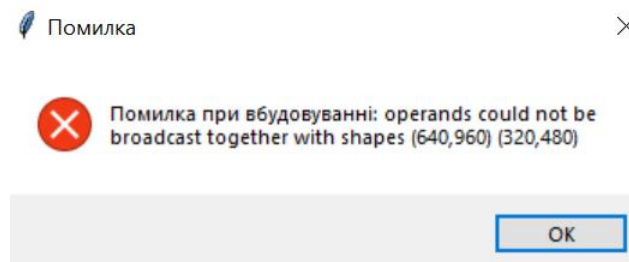


Рисунок 3.3 – Помилка при завантаженні різних зображень за розміром у методі DWT

В цьому методі відбувається ключова валідація операцій, зокрема: чи коректно визначено метод додавання водяного знаку, перевіряється виклик необхідної функції обробки (`dct_embed()` або `dwt_embed()`), а також здійснюється показ результатів та обчислення метрик PSNR і SSIM (рис.3.4), які оцінюють якість зображення. Важливо відзначити, що впроваджено перевірку обробки виняткових ситуацій в програмі за допомогою блоків (try-ехсепт) [20].

Метод	Альфа	Атака	PSNR	SSIM
DWT	1	без атаки	23.55	0.7925
DWT	1	шум	20.82	0.4142
DWT	1	обрізання	10.17	0.1006
DWT	1	стиснення	22.56	0.6292

Рисунок 3.4 – Обчислення метрик PSNR та SSIM

Нижче подається опис двох ключових способів вкладання водяних знаків: з використанням дискретного косинусного перетворення (ДКП) та вейвлет-перетворення, з врахуванням вагового коефіцієнта α .

```
def dct_embed(self):
    watermark_resized = cv2.resize(self.watermark_image,
                                    (self.original_image.shape[1], self.original_image.shape[0]))

    original_dct = cv2.dct(np.float32(self.original_image) / 255.0)
    watermark_dct = cv2.dct(np.float32(watermark_resized) / 255.0)

    watermarked_dct = original_dct + self.alpha * watermark_dct
    self.watermarked_image = np.uint8(cv2.idct(watermarked_dct) * 255)

def dwt_embed(self):
    watermark_resized = cv2.resize(self.watermark_image,
                                    (self.original_image.shape[1] // 2, self.original_image.shape[0] // 2))

    coeffs_original = pywt.dwt2(self.original_image, 'haar')
    cA, (cH, cV, cD) = coeffs_original

    coeffs_watermark = pywt.dwt2(watermark_resized, 'haar')
    cA_w, (cH_w, cV_w, cD_w) = coeffs_watermark

    cH_watermarked = cH + self.alpha * cH_w
    coeffs_watermarked = (cA, (cH_watermarked, cV, cD))
    self.watermarked_image = np.uint8(pywt.idwt2(coeffs_watermarked, 'haar'))
```

За допомогою методу `dct_embed(self)` впроваджується водяний знак у зображення, використовуючи дискретне косинусне перетворення (DCT). Алгоритм працює наступним чином: спочатку масштабує зображення водяного знаку до розмірів оригіналу в пікселях, після чого обчислює DCT для обох зображень. Далі, використовуючи визначений ваговий коефіцієнт α , алгоритм накладає межу інтенсивності та впливу (прозорості) на зображення.

Завершується процес обчисленням зворотного DCT для отримання кінцевого результату.

Робота методу вейвлет-перетворень DWT полягає в тому, щоб накласти водяний знак на зображення, використовуючи метод DWT. Також метод вирівнює та масштабує водяний знак відповідно до оригіналу, після чого обчислює DWT обох зображень, щоб отримати складові: cA, cH, cV, cD. Потім відбувається вбудовування знаку у горизонтальну складову cH, і на завершення виконується зворотне DWT для створення зображення з водяним знаком.\

```
def calculate_metrics(self, img1, img2):
    if img1.shape != img2.shape:
        img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))

    psnr_val = psnr(img1, img2)
    ssim_val = ssim(img1, img2)
    return psnr_val, ssim_val

def apply_attack(self, attack_type):
    if self.watermarked_image is None:
        messagebox.showwarning("Увага", "Спочатку вбудовуйте водяний знак")
        return

    try:
        attacked = self.watermarked_image.copy()

        if attack_type == "шум":
            noise = np.random.normal(0, 25, self.watermarked_image.shape)
            attacked = np.clip(attacked + noise, 0, 255).astype(np.uint8)
            self.status_var.set("Застосовано атаку: Гаусівський шум")

        elif attack_type == "обрізання":
            h, w = attacked.shape
            crop_h, crop_w = int(h * 0.1), int(w * 0.1)
            attacked = attacked[crop_h:-crop_h, crop_w:-crop_w]
```

```

        attacked = cv2.resize(attacked, (w, h))
        self.status_var.set("Застосовано атаку: Обрізання 10%")

elif attack_type == "стиснення":
    encode_param = [int(cv2.IMWRITE_JPEG_QUALITY), 50]
    _, encimg = cv2.imencode('.jpg', attacked, encode_param)
    attacked = cv2.imdecode(encimg, cv2.IMREAD_GRAYSCALE)
    self.status_var.set("Застосовано атаку: JPEG стиснення (50%)")

elif attack_type == "розмиття":
    attacked = cv2.GaussianBlur(attacked, (5, 5), 0)
    self.status_var.set("Застосовано атаку: Розмиття Гауса 5x5")

self.watermarked_image = attacked
self.update_display("Результат", self.watermarked_image)

except Exception as e:
    messagebox.showerror("Помилка", f"Помилка при атаці: {e}")

```

Застосовані вище способи імітують атаки на зображення, серед них: додавання шуму, обрізання, стиснення та розмиття. Параметри, що були використані для моделювання атак, включають: додавання Гаусівського шуму, обрізання 10% країв зображення (якщо водяний знак розташований по краях), стиснення зображення у форматі JPEG з рівнем якості 50%, а також розмиття за допомогою ядра Гауса 5x5 (рис.3.5).

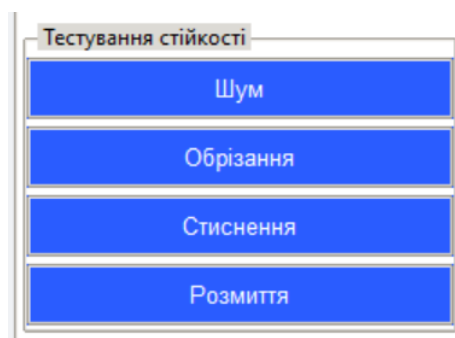


Рисунок 3.5 – Вікно тестування стійкості зображення

У межах розробки програмного забезпечення для системи захисту цифрових зображень була реалізована функціональність оцінювання якості зображень після деструктивного впливу зовнішніх атак. Зокрема, реалізовано метод `calculate_metrics(self, img1, img2)`, який дозволяє здійснювати обчислення метричних характеристик, таких як PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index Measure).

Ці метрики забезпечують кількісну оцінку збереження візуальної якості вбудованого водяного знаку після застосування атак (наприклад, шуму, обрізання, стиснення або розмиття). У разі невідповідності розмірів оригінального та обробленого зображень передбачено автоматичне масштабування водяного знаку до розміру вихідного зображення, що дозволяє коректно виконати порівняльний аналіз. [18]

У результаті виконаної роботи було створено функціональний програмний застосунок, орієнтований на впровадження механізмів захисту цифрових зображень з використанням методів частотного аналізу, зокрема дискретного косинусного (DCT) та вейвлет-перетворення (DWT).

Програма містить інтуїтивно зрозумілий графічний інтерфейс користувача, а також повний набір інструментів для вбудовування, витягання та аналізу цифрових водяних знаків.

Особливістю розробленого інтерфейсу є інтегрована система тестування на стійкість до атак, яка включає обчислення показників PSNR та SSIM, що дозволяє проводити експериментальне дослідження стійкості алгоритмів стеганографічного вбудовування. Таким чином, програмний засіб не лише забезпечує базову функціональність захисту, але й виконує роль дослідницького інструменту для подальшої оптимізації алгоритмів стійкої стеганографії.

3.3 Тестування розробленої системи захисту та аналіз його роботи

Проведене тестування є видом функціонального тестування, що зосереджується на перевірці коректності втілення ключових функцій системи захисту цифрових водяних знаків, включаючи процеси накладання, вилучення, збереження, обробки та протидії атакам. Завдання тестування – оцінити правильність роботи складових системи за звичних умов експлуатації та перевірити відповідність отриманих результатів запланованим параметрам (якість, стабільність, точність).

У даному випадку було написано скрипт за допомогою якого відбувається автоматичне тестування, перевірка основних методів та модулів програмного забезпечення, параметр α , виведення точності схожості зображень після накладання водяного знаку. Перевірки оцінки модулів, було здійснено з параметром α з інтенсивністю у 10 як для методу DCT та DWT.

Вхідні зображення які було використано для тестування системи захисту продемонстровано на (рис. 3.6).



Рисунок 3.6 – Вхідні дані для тестування різних форматів DCT

Зображення було взято різного формату оригінал у форматі JPEG та водяний знак у форматі PNG. Важливо перевірити різний вид форматів зображення так як це займає тривалий час для користувача, але програма дозволяє робити це автоматично.

Перевірено й те, як саме програма видаляє водяний знак з картинки, та чи зберігає вона вихідне зображення без змін після цих дій. На рисунку 3.7 показано процес накладання та зняття водяного знаку, застосовуючи дискретне косинусне перетворення.

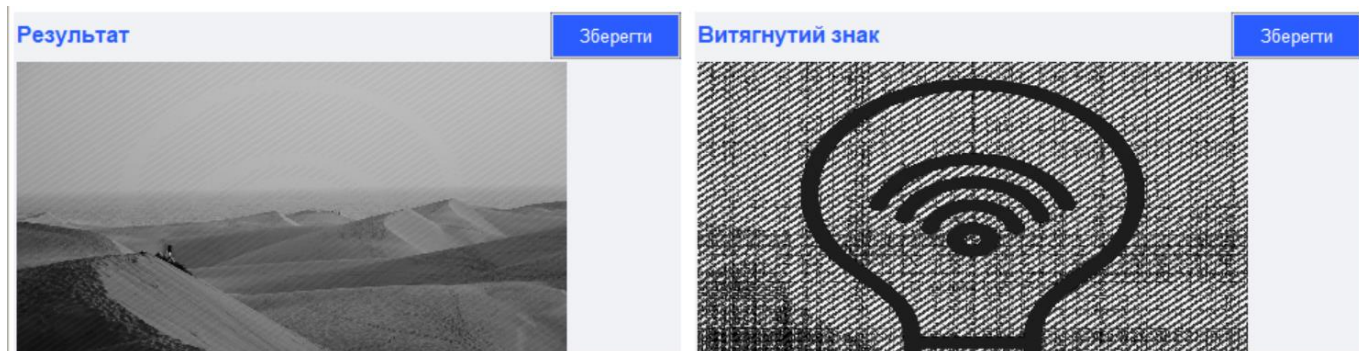


Рисунок 3.7 – Тестування накладання та витягування водяного знаку методом DCT

Далі продемонстровано тестування методу DWT- методу вейвлет-перетворень на чіткість накладання та витягу водяного знаку на оригінальне зображення (рис. 3.8). Вхідними даними тестування системи захисту були 2 однакові зображення, для того, щоб протестувати чіткість витягу водяного знаку.



Рисунок 3.8 – Тестування чіткості накладання та витяг водяного знаку методом DWT

Результатом тестування на накладання водяного знаку є зображення яке є непошкодженим, має невелику різницю між оригінальним зображенням. На (рис.3.7-3.8), бачимо що система працює як з зображеннями різного програмного формату та розмірного, та працює стабільно. Наступним кроком буде перевірка на стійкість зображення до атак які здійснюються для присвоєння власності автора. Найпопулярнішими з них є стиснення, обрізання, накладання шуму (рис.3.9) та розмиття зображення, щоб приховати водяний знак.



Рисунок 3.9 – Накладання шуму на захищене зображення DCT

Після накладання шуму Гауса на захищене зображення методом дискретного косинусного перетворення (DCT), бачимо, що водяний знак залишається на зображенні, та його легко можна ідентифікувати. Тобто можна сказати, що накладання шуму не може замаскувати накладений водяний знак методом DCT [20].

У наступному етапі відбувається перевірка захищеного зображення на обрізання зображення після виконаного дискретного косинусного перетворення зображено на (рис 3.10), якщо візуально можна буде розгледіти водяний знак з невеликою інтенсивністю α , то тестування даного етапу буде пройдено.

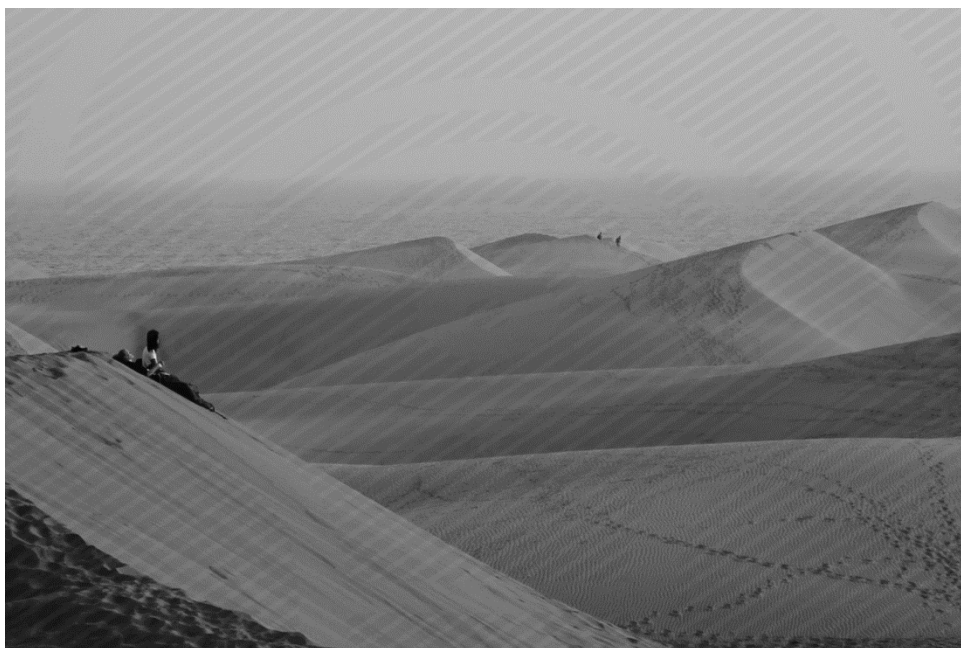


Рисунок 3.10 – Накладання обрізання на захищене зображення DCT

Навіть при невеликому показнику α , тобто силі втручання водяного знаку у оригінальне зображення, візуально видно, що водяний знак залишається наявним на зображенні, та його легко можна ідентифікувати. Тестування даного методу атаки вважається пройденим. Перевіряємо наступний метод імітації атаки на зображення під назвою стиснення, після перетворення на (рис 3.11), параметр α залишається за замовчуванням.



Рисунок 3.11 – Накладання стиснення на захищене зображення DCT

У результаті проходження тестування стиснення захищеного зображення, наглядно бачимо наявний водяний знак на зображенні. Отже, можна сказати, що даний етап тестування системи захисту зображень пройдено успішно.

Заключним методом тестування на вразливість до атак захищеного зображення, буде розмиття за допомогою ядра Гауса 5×5 зображеного на (рис 3.12).



Рисунок 3.12 – Накладання розмиття на захищене зображення DST

Під час тестування даним методом було використано 3 разове накладання ефекту розмиття, для того, щоб остаточно перевірити як вплине такий вид атаки на захищене зображення, результат виявився позитивним. Через візуально наявний водяний знак на зображенні, можна сміливо сказати, що тестування на атаки система захисту зображень методом дискретного косинусного перетворення пройшла успішно.

3.4 Експериментальне порівняльне тестування стійкості цифрового водяного знака

Експериментальне тестування - це спосіб перевірки функціональності програмного забезпечення, коли тестувальник досліджує роботу системи, не маючи доступу до її коду чи документації. В даному дослідженні було проведено експериментальне випробування розробленої системи цифрового водяного знакування, втіленої у вигляді програмного модуля, написаного на мові Python.

Тестування виконано з використанням автоматизованого коду, що забезпечує почергове впровадження водяного знака, імітацію поширених атак на зображення, а також наступне вилучення знака з оцінкою його якості. Цей підхід дозволяє ретельно вивчити реакцію системи в різноманітних умовах та кількісно визначити її продуктивність.

Код забезпечує пакетне тестування, або batch testing: для кожного з заздалегідь визначених значень параметра вбудовування ($\alpha = 1, 5, 10, 20$) здійснюється серія експериментів з водяним знаком, застосовуючи до нього різноманітні атаки (без атаки, додавання шуму, обрізання, стиснення JPEG, розмиття). Під час кожного експерименту автоматично обчислюються значення показників PSNR та SSIM, котрі широко визнані метриками оцінки якості зображень [18].

У таблиці 3.4 наведено отримані результати після проходження експериментально тестування методу дискретного косинусного перетворення при таких параметрах як, різний коефіцієнт α від 1 до 20, для різного виду атак на зображення. Проведення такого типу тестування дозволить зрозуміти, у вигляді різних метрик, таких як PSNR та SSIM хорошим показником вважається >20 dB.

Таблиця 3.4 – Тестування методу DCT

Метод	Альфа	Атака	PSNR	SSIM
DCT	1	Без атаки	40.68	0.9961
DCT	1	Шум	10.82	0.1602
DCT	1	Обрізання	10.01	0.0467
DCT	1	Стиснення	19.41	0.6903
DCT	1	Розмиття	20.25	0.4792
DCT	5	Без атаки	9.4	0.3305
DCT	5	Шум	9.32	0.2041
DCT	5	Обрізання	10.15	0.1893
DCT	5	Стиснення	9.48	0.3251
DCT	5	Розмиття	11.3	0.3273
DCT	10	Без атаки	9.06	0.1936
DCT	10	Шум	8.9	0.1723
DCT	10	Обрізання	10.27	0.1926
DCT	10	Стиснення	8.99	0.1899
DCT	10	Розмиття	11.14	0.2639
DCT	20	Без атаки	9.11	0.1598
DCT	20	Шум	8.97	0.1398
DCT	20	Обрізання	10.26	0.17
DCT	20	Стиснення	9.04	0.1436
DCT	20	Розмиття	11.15	0.1977

В результаті проведеного дослідження з використанням експериментів було з'ясовано доцільність застосування методу дискретного косинусного перетворення (ДКП) для нанесення цифрових водяних знаків. Ефективність методу підтверджена успішним видаленням вбудованого знака за відсутності будь-яких перешкод, про що свідчать високі показники обчислених метрик PSNR та SSIM. Отримані дані вказують на збереження якості початкового зображення та самого водяного знака, а також на мінімальні зміни вихідного контенту під час вбудовування.

Перейдемо до детального розгляду результатів, поданих на графіках, які демонструють характеристики методу дискретного косинусного перетворення (DCT) в різних умовах (рис. 3.13-3.14). Це дає змогу візуально оцінити зміни якості видобутого водяного знаку при зміні параметрів вбудовування та впровадженні стандартних атак, що сприятиме обґрунтованим висновкам щодо продуктивності методу.

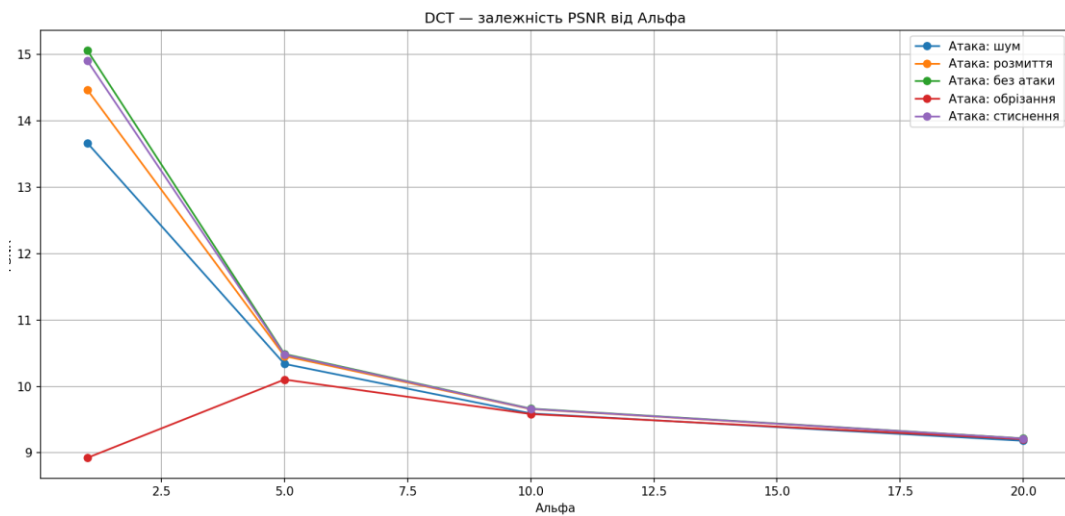


Рисунок 3.13 – Графік залежності PSNR від альфа на методі DCT

Також побудовано графік залежності метрик SSIM від коефіцієнта alpha для методу дискретного косинусного перетворення, продемонстровано на рисунку 3.14.

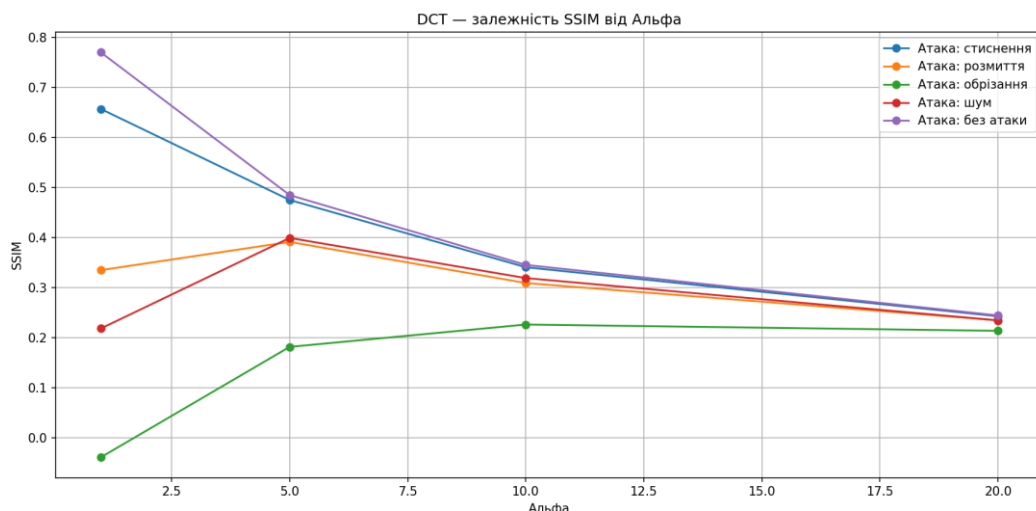


Рисунок 3.14 – Графік залежності SSIM від альфа на методі DCT

Після проходження тестування методу DCT можна сказати, що Отже, метод DCT здатний надати прийнятний компроміс між збереженням якості зображення та надійним витягуванням водяного знака, якщо раціонально обрати параметри. Він вирізняється легкістю в реалізації, високою швидкістю обчислень та має потенціал для застосування в реальних ситуаціях, де можливі ушкодження зображення внаслідок обробки або передачі. Для посилення захисту від серйозних атак, розумно розглянути комбіновані стратегії, засновані на DCT [21].

Наступне тестування буде проведено для методу DWT, для того, щоб оцінити даний метод у точності накладання та усунення водяного знаку з зображення, та порівняти результати з попереднім методом косинусного перетворення. Точність метрик від залежності коефіцієнта α показано у таблиці 3.5.

Таблиця 3.5 – Тестування методу DWT

Метод	Альфа	Атака	PSNR	SSIM
DWT	1	Без атаки	23.55	0.7925
DWT	1	Шум	20.83	0.4164
DWT	1	Обрізання	10.17	0.1006
DWT	1	Стиснення	22.56	0.6292
DWT	1	Розмиття	8.98	0.0491
DWT	5	Без атаки	10.43	0.3658
DWT	5	Шум	10.37	0.3284
DWT	5	Обрізання	9.13	0.1017
DWT	5	Стиснення	10.42	0.3514
DWT	5	Розмиття	8.81	0.0157
DWT	10	Без атаки	9.57	0.2373
DWT	10	Шум	9.54	0.2182
DWT	10	Обрізання	8.95	0.0777
DWT	10	Стиснення	9.56	0.2309
DWT	10	Розмиття	8.79	0.2639

Продовження таблиці 3.5

DWT	20	Без атаки	9.15	0.1354
DWT	20	Шум	9.13	0.1239
DWT	20	Обрізання	8.85	0.0457
DWT	20	Стиснення	9.14	0.1307
DWT	20	Розмиття	8.78	0.0114

За результатами експериментального випробування методу дискретного вейвлет-перетворення (DWT) в контексті цифрового водяного знакування, можна виокремити ряд важливих спостережень стосовно його продуктивності та здатності витримувати вплив зовнішніх факторів. DWT продемонстрував високу ефективність у збереженні якості вбудованого зображення, зокрема, при невисоких значеннях параметра α . Водночас, точність видобування водяного знака залишається на достатньому рівні навіть після незначних атак, що вказує на позитивну локальну стійкість вейвлет-компонент до спотворень.

Переходимо до розгляду графічного матеріалу, що демонструє результати експериментального дослідження ефективності методу дискретного вейвлет-перетворення (DWT). Створені графіки дозволяють візуально спостерігати, як змінюється якість видобутого водяного знаку залежно від різних атак та при зміні параметру вбудовування α . (рис 3.15-3.16).

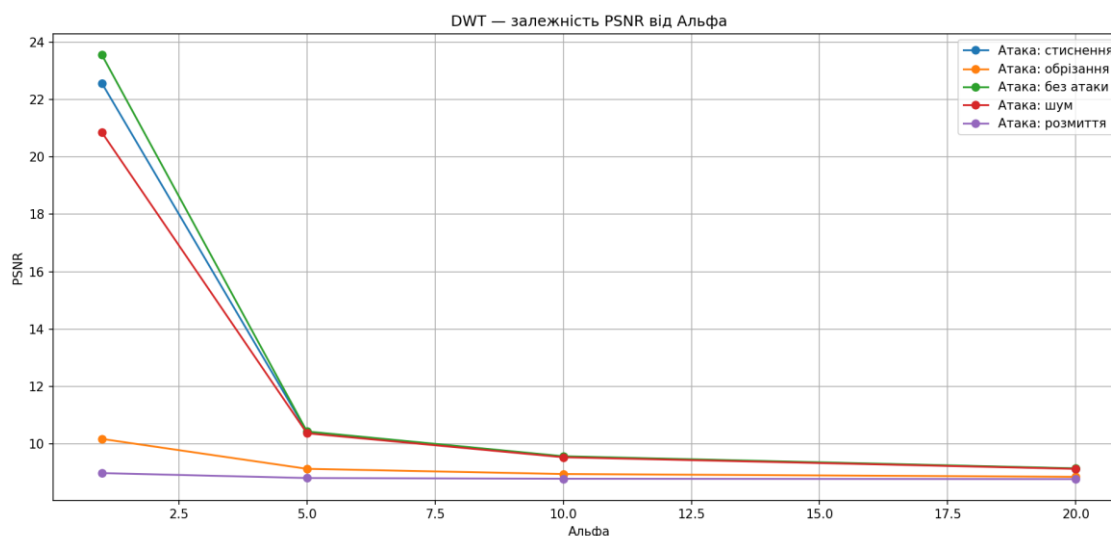
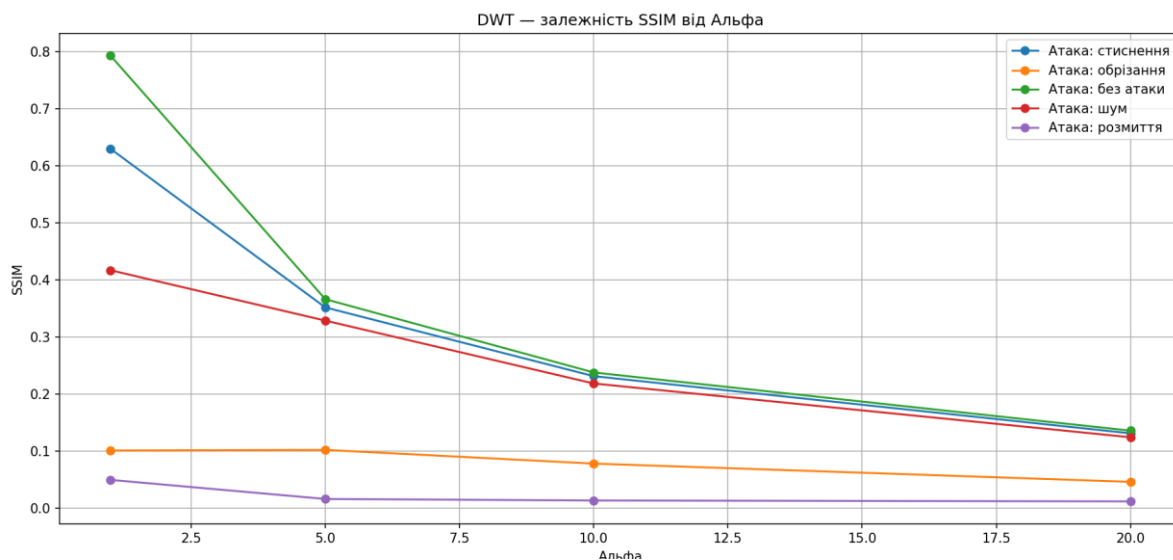


Рисунок 3.15 – Графік залежності PSNR від альфа на DWT

Нижче представлено графік, що демонструє взаємозв'язок між показником SSIM та коефіцієнтом α для методу вейвлет-перетворень. Дані для побудови графіка було отримано безпосередньо з модуля експериментального тестування, що проілюстровано на рисунку 3.16.



Рисунк 3.16 - Графік залежності SSIM від альфа на методі DWT

На основі отриманих даних експериментального дослідження методів цифрового водяного знакування, що базуються на дискретному косинусному перетворенні (DCT) та дискретному вейвлет-перетворенні (DWT), стає можливим зробити узагальнюючі висновки стосовно їхньої продуктивності та здатності протистояти зовнішнім впливам. Обидва методи гарантують високий рівень вбудовування та точність вилучення водяного знака за відсутності атак, що підтверджується високими показниками PSNR та SSIM [21].

В умовах застосування стандартних атак, зокрема, додання шуму, стиснення JPEG, обрізання або розмиття, обидва методи виявляють різну поведінку. DCT демонструє більшу чутливість до просторових деформацій, тоді як DWT характеризується кращою здатністю зберігати водяний знак при локалізованих змінах зображення або при компресії. Це свідчить про відмінні механізми стійкості в кожному підході.

Параметр α , що визначає силу вбудовування, відіграє важливу роль у взаємозв'язку між візуальною якістю зображення та збереженням водяного знака. Збільшення цього параметра веде до покращення стійкості до атак, проте одночасно призводить до збільшення спотворень в зображенні, на яке накладено водяний знак.

Загалом, обидва методи показали перспективність у застосуванні для цифрового захисту зображень. DWT продемонстрував дещо вищу стійкість під час атак, водночас DCT вирізняється простотою реалізації та високою швидкістю обчислень. Відповідно до поставлених завдань, може бути доцільним використання кожного з методів окремо або їх поєднання в системах цифрового водяного маркування.

4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

4.1. Порівняння з аналогічними системами захисту

Оцінка ефективності запропонованої системи цифрового водяного знакування неможлива без її порівняння з уже існуючими методами та технологіями, які використовуються як у наукових дослідженнях, так і в комерційних продуктах. До найпоширеніших підходів належать методи, що базуються на маніпуляціях з найменш значущими бітами (LSB), фазові методи, а також гібридні техніки, які поєднують у собі властивості різних перетворень, таких як дискретне косинусне перетворення (DCT), дискретне вейвлет-перетворення (DWT), сингулярне розкладення матриці (SVD) тощо. Кожен із зазначених методів має свої переваги і недоліки з точки зору стійкості до атак, збереження якості зображення та обчислювальної складності [22].

Зокрема, LSB-методи є простими у реалізації та малозатратними з точки зору обчислювальних ресурсів, проте вони мають низьку стійкість до атак, таких як компресія або додавання шуму. Фазові методи та методи, засновані на перетвореннях (наприклад, DCT або DWT), демонструють вищу стійкість, оскільки дозволяють вбудовувати водяний знак у менш вразливі до спотворень частини зображення. Гібридні підходи, які поєднують переваги кількох методів, дозволяють досягти оптимального балансу між якістю зображення, стійкістю до атак і складністю реалізації.

Комерційні рішення, такі як Digimarc або SigniFlow, застосовують багаторівневі алгоритми з акцентом на високий рівень захисту авторських прав та юридичну валідність. Ці системи зазвичай є закритими за структурою, мають складну архітектуру та орієнтовані на масштабне використання в корпоративному секторі. Вони забезпечують надійне вбудовування водяних знаків, їх виявлення після активних і пасивних атак, а також підтримку правових процедур у випадку порушення інтелектуальної власності.

У таблиці 4.1 показано методи накладання водяного знаку на зображення

Таблиця 4.1 – Порівняльні методи накладання водяного знаку

Метод	Стійкість до атак	Якість зображення	Складність реалізації	Застосування
LSB	Низька	Висока	Низька	Просте хобі, навчання
DCT	Середньо-висока	Середня	Середня	Компресовані зображення
DWT	Висока	Висока	Висока	Різні масштаби атак
SVD	Висока	Середня	Висока	Авторське право
DWT+DCT або SWD	Дуже висока	Висока	Висока	Комерційні продукти

У порівнянні з вказаними комерційними продуктами, запропоноване програмне забезпечення є більш відкритим, адаптивним і зручним для експериментального використання. Воно дозволяє досліднику або розробнику обирати між кількома алгоритмічними підходами, оцінювати якість і стійкість водяного знаку шляхом візуалізації результатів, а також модифікувати систему під специфічні вимоги [23]. Така гнучкість робить систему особливо корисною для академічного середовища, дослідницьких лабораторій, а також для локальних або навчальних проєктів, де важливою є не лише ефективність, але й зрозумілість внутрішньої логіки функціонування алгоритмів.

4.2. Переваги та обмеження реалізованого підходу

Запропонована система цифрового водяного знакування вирізняється кількома вагомими перевагами, що робить її доцільною як для дослідницьких цілей, так і для практичного застосування. Серед ключових позитивних сторін варто відзначити гнучкість вибору між двома популярними частотними перетвореннями – дискретним косинусним (DCT) (рис. 4.1) та дискретним вейвлетним (DWT). Така можливість вибору дозволяє пристосувати систему до

конкретного типу зображень або до потенційних загроз. Скажімо, DCT добре себе показав у контексті стійкості до JPEG-компресії та додаткового шуму, в той час як DWT є більш ефективним проти розмиття чи фрагментації (обрізання).



Рисунок 4.1 – Зображення до та після вбудовування водяного знаку

Не менш важливою перевагою є впровадження графічного інтерфейсу користувача, який забезпечує інтуїтивне використання програмного забезпечення. Це значно розширює коло потенційних користувачів, роблячи систему доступною не лише фахівцям з обробки зображень, а й користувачам без спеціальних знань. Візуалізація результатів вставки та вилучення водяного знаку, разом із інтерактивним налаштуванням параметрів, зокрема коефіцієнта вставки (α), сприяють кращому розумінню механізмів водяного знакування та підвищують ефективність експериментів [23].

Разом з тим, як і будь-яка технічна розробка, запропонований підхід має певні обмеження. Передусім, система демонструє знижену ефективність при комбінованих атаках, коли зображення з водяним знаком піддається одночасно кільком типам спотворень (наприклад, шум+розмиття, JPEG+обрізання). Також варто враховувати вплив характеристик самого зображення: рівень

текстурованості, контрастність і яскравість можуть суттєво впливати на якість вбудовування та точність витягання водяного знаку.

Особливе значення в роботі системи має параметр α , який визначає інтенсивність вставки водяного знаку. Збільшення цього параметра дійсно покращує стійкість до атак, але може погіршити візуальну якість зображення, що може бути неприйнятним у деяких задачах, особливо коли зображення планується використовувати або публікувати.

4.3 Перспективи розвитку

Подальший розвиток системи цифрового водяного знакування передбачає розширення її функціональних можливостей, підвищення стійкості до атак та адаптацію до новітніх вимог захисту цифрового контенту. Одним із найперспективніших напрямів є впровадження гібридних методів вбудовування, зокрема комбінацій DWT+DCT або DWT+SVD. Такі підходи дозволяють поєднувати переваги різних перетворень: наприклад, дискретне вейвлет-перетворення (DWT) забезпечує просторово-частотну локалізацію ключових характеристик зображення, тоді як DCT або SVD дозволяють здійснювати стійке та малопомітне вбудовування водяного знаку в трансформовані коефіцієнти. Використання таких комбінованих схем значно підвищує стійкість системи до складних атак, включаючи комбіновані або геометричні спотворення [24].

Іншим актуальним вектором розвитку є інтеграція з блокчейн-технологіями. Завдяки децентралізованій природі блокчейн-платформ, можлива фіксація фактів авторства, часу створення зображення або внесення змін у публічному незмінному реєстрі. Це створює додатковий рівень надійності та правового захисту для авторів цифрового контенту. Зокрема, використання смарт-контрактів дозволяє автоматизувати перевірку автентичності та прав на зображення.

У цьому ж контексті важливо відзначити перспективу інтеграції з технологією NFT (Non-Fungible Tokens). Поєднання цифрового водяного

знакування з токенизацією зображень дозволяє не лише підтвердити авторство, але й створити унікальні цифрові активи, які можуть бути куплені, продані або передані з гарантією автентичності.

Окремий напрям модернізації системи стосується застосування методів машинного навчання для автоматичної адаптації параметрів водяного знакування. Зокрема, використання глибоких нейронних мереж може дозволити системі аналізувати статистичні характеристики зображення і визначати оптимальні області для вставки водяного знаку, що підвищить як стійкість до атак, так і зменшить вплив на візуальну якість. Крім того, моделі навчання можуть бути використані для розпізнавання типів атак та адаптації методу вилучення знаку у відповідь на виявлені спотворення.

Ще одним перспективним етапом розвитку є розширення функціоналу до роботи з відеоданими. У цьому випадку водяний знак може вбудовуватися не лише у просторову частину окремих кадрів (наприклад, ключових), а й у часову компоненту відеопотоку. Це дозволить створювати захищений мультимедійний контент з вбудованими маркерами, які залишаються стійкими до трансформацій, таких як перекодування, обрізання або зміна частоти кадрів.

4.4. Варіанти комерціалізації та практичного застосування

Розроблене програмне забезпечення цифрового водяного знакування має значний потенціал для практичного використання у різноманітних галузях, де актуальними є питання захисту авторських прав, підтвердження автентичності цифрових матеріалів та боротьби з фальсифікацією контенту. Однією з найбільш очевидних сфер застосування є захист цифрового мистецтва, фотографій, графічного дизайну, а також відео- та мультимедійного контенту, що створюється індивідуальними авторами або креативними студіями [25]. В умовах масового поширення контенту в соціальних мережах і онлайн-платформах, вбудований цифровий водяний знак може стати інструментом не лише захисту авторства, а й відстеження поширення зображень.

Крім цього, система може знайти застосування в медичній сфері, зокрема для верифікації достовірності та джерела медичних зображень (наприклад, рентгенограм, МРТ або КТ-знімків), де критично важливим є збереження цілісності даних. Аналогічно, у сфері журналістики та розслідувань, цифровий водяний знак може виступати як механізм підтвердження автентичності відеоматеріалів, що особливо актуально в умовах інформаційних війн і масових дезінформаційних кампаній.

З точки зору комерціалізації, існує кілька перспективних форматів:

1. Desktop-додаток з розширеним функціоналом, орієнтований на професійних користувачів – дизайнерів, фотографів, відеоредакторів, які потребують локального, надійного та швидкого інструменту захисту контенту. Такий застосунок може включати модулі пакетної обробки, налаштування стійкості, візуалізації результатів та звітування.

2. REST API або SDK – програмний інтерфейс для інтеграції з іншими системами, зокрема з вебплатформами (наприклад, фотостоками, CMS, відеохостингами). Це дозволяє розширити функціонал існуючих сервісів автоматичним захистом контенту на стороні сервера, що є особливо актуальним для B2B-сегменту.

3. SaaS-рішення (Software as a Service) – хмарна платформа, яка надає доступ до функціоналу водяного знакування через браузерний інтерфейс або API. Це дає можливість масштабування сервісу, адаптації під різні моделі підписки та обслуговування широкого кола користувачів без потреби у встановленні локального ПЗ.

4. Мобільні застосунки – адаптація системи під iOS та Android відкриває перспективи для використання у сфері мобільної безпеки, зокрема для швидкого додавання водяного знаку до фото- і відеоконтенту безпосередньо з камери смартфона. Така функціональність особливо корисна для журналістів, блогерів, туристів, а також користувачів NFT-майданчиків [26].

Розглянемо таблицю переваг та недоліків можливостей бізнес моделей які можна розглянути для майбутнього розвитку даної системи захисту.

Таблиця 4.2 – Таблиця можливостей бізнес-моделей

Модель	Цільова аудиторія	Переваги	Недоліки
Desktop-додаток	Дизайнери, фотографи	Локальне використання, приватність	Не масштабовано
REST API	Вебплатформи, стартапи	Гнучка інтеграція	Потреба у серверній частині
SaaS	Масовий ринок	Доступність, масштабованість	Підписка, залежність від хмари
Мобільний додаток	Журналісти, блогери	Зручність, мобільність	Обмежена обчислювальна потужність

Таким чином, гнучкість архітектури розробленого ПЗ, підтримка сучасних алгоритмів і можливість масштабування дозволяють реалізувати різні моделі комерціалізації – від інструментів для індивідуального використання до інтегрованих B2B-рішень [27]. У сукупності це формує стійке підґрунтя для подальшої трансформації наукової розробки в конкурентоспроможний ринковий продукт.

ВИСНОВКИ

В бакалаврській кваліфікаційній роботі було розроблено систему захисту зображень, використовуючи методи частотних перетворень. Досягнуто вагомих результатів у дослідженні застосунку, що надасть можливість авторам захистити свої зображення від неправомірних дій злоумисників, які порушують авторські права, копіюючи роботи без згоди творця. Основними інструментами для реалізації цієї системи стали бібліотеки OpenCV, SciPy, PyWavelets та NumPy, що були інтегровані задля створення необхідного застосунку. Ці бібліотеки дозволили розробити програмне забезпечення завдяки своїй зручності у використанні та гнучкості інструментів, що містяться в них, сприяючи отриманню певних результатів у нанесенні водяних знаків на зображення різних форматів.

Одним із головних помічників, що використовувався під час впровадження системи цифрового водяного знака, стала мова програмування Python. Вона надала чималий функціонал для роботи з сторонніми бібліотеками, побудови графічного інтерфейсу, створення основного програмного блоку, а також для проведення дослідних випробувань алгоритмів. Через гнучкий синтаксис, потужну екосистему наукових та візуалізаційних бібліотек, таких як OpenCV, PyWavelets, Matplotlib та інші, Python дав можливість ефективно втілити необхідні етапи розробки, оцінки якості та візуалізації здобутків у комфортному та масштабованому форматі.

Розроблене програмне забезпечення пройшло ретельне тестування з використанням метрик структурної подібності (SSIM) та пікового відношення сигнал/шум (PSNR), що входять до ключових показників якості в області обробки зображень. Експерименти засвідчили ефективність втілених алгоритмів – результати продемонстрували високі показники вибраних метрик, що вказує на точне вилучення водяного знака та мінімальну втрату якості зображення після вбудовування. Програма показала стабільну роботу навіть під впливом типових атак, таких як адитивний шум, гаусове розмиття, JPEG-стиснення та обрізка

фрагментів зображення. Ці атаки часто застосовуються зловмисниками для спотворення або приховування цифрових авторських маркерів. Попри ці зовнішні впливи, система цифрового водяного знакування зберегла здатність розпізнавати водяний знак, що підтверджує її надійність, стійкість та практичну застосовність у контексті захисту цифрового контенту від несанкціонованого копіювання.

Розробка цієї системи засвідчила значний потенціал програми для подальшого вдосконалення. Зокрема, існує можливість розширення функціоналу шляхом додавання нових типів актуальних атак, що використовуються у практиці цифрового піратства, а також впровадження альтернативних методів вбудовування водяного знака. Проведене дослідження підтвердило, що завдяки інтуїтивно зрозумілому та зручному графічному інтерфейсу користувач має змогу легко взаємодіяти з програмним забезпеченням, ефективно захищаючи власні цифрові матеріали від несанкціонованого копіювання. Таким чином, створена система може стати надійним інструментом для авторів і розробників, які прагнуть убезпечити свої роботи від зловмисного використання.

Усі поставлені задачі в рамках проекту було виконано з успіхом, що підтверджується великою кількістю експериментальних даних, продемонстрованих у вигляді графіків, таблиць та візуалізацій. Впроваджена система цифрового захисту зображень слугує яскравим прикладом дієвого використання інструментів комп'ютерної автоматизації для вирішення нагальних проблем, пов'язаних з охороною авторських прав та збереженням цілісності візуального контенту.

Створений додаток не тільки виконує основні функції водяного знакування, але й демонструє високу точність та надійність при тестуванні у різних середовищах, зокрема, під впливом типових атак. Це свідчить про його практичну користь та широкі можливості для майбутнього вдосконалення, наприклад, через інтеграцію найновіших методів стеганографії, покращення алгоритмів адаптації до атак та розширення функціоналу графічного інтерфейсу для ще більшої зручності користування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конахович Г.Ф. Комп'ютерна стеганографія. Теорія і практика / Г.Ф. Конахович, А. Ю. Пузиренко. – Київ: МК-Пресс, 2006. – 288 с. URL: <https://repository.hneu.edu.ua/bitstream/123456789/2289/1/Стеганографія.pdf> (дата звернення 19.05.2025)
2. Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп'ютерної стеганографії: Навч. посіб. для студентів і аспірантів. —Вінниця: ВДТУ, 2003 – 50-53с. URL: https://pdf.lib.vntu.edu.ua/books/2018/Xoroshko_Komputer_2017_155.pdf (дата звернення 19.05.2025)
3. N.F. Johnson, S. Jajodia, Steganalysis: The Investigation of HiddenInformation, IEEE Information Technology Conference, Syracuse, New York,USA, Sept. 1st-3rd. 1998. URL: https://cse.buffalo.edu/courses/cse725/peter/Johnson_1997.pdf (дата звернення 19.05.2025)
4. Юдін О.К., Зюбіна Р.В., Фролов О.В. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів. - Pedram // 31st International Conference on Distributed Computing Systems Workshops. Minneapolis, USA, 2011. P. 1-6. URL: <https://journals.indexcopernicus.com/api/file/viewByFileId/964313> (дата звернення 19.05.2025)
5. Lecture Notes on Cryptography / Goldwasser S., Bellare M. – Cambridge, Massachusetts, 2001. – 283 p. URL: <https://do.csc.knu.ua/wpcontent/uploads/2023/01/gb.pdf> (дата звернення 19.05.2025)
6. Лісковський І.О., Кулик М.В. Реалізація стеганографічної системи для відео з використанням сингулярного розкладання. // Дванадцята міжнародна науково-технічна конференція «Проблеми телекомунікацій»; Десята міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем»: Матеріали

конференції. К.: НТУУ "КПІ", 2018. – с. 288-290. URL: <https://conferenc-journal.its.kpi.ua/article/view/131540/127619> (дата звернення 19.05.2025)

7. Darmstaedter V., Delaigle J.-F., Quisquater J., Macq B. Low cost spatial watermarking // Computers and Graphics. 1998. Vol. 5. P. 417 – 423. URL: <https://home.cis.rit.edu/~cnspci/references/darmstaedter1998.pdf> (дата звернення 19.05.2025)

8. Задірака В. К. Аналіз стійкості стеганографічних систем в моделі пасивного противника / В. К. Задірака, Н. В. Кошкіна, О. С. Олексюк // Штучний інтелект. — 2004. — № 3. — С. 801—805. URL: http://iai.dn.ua/public/JournalAI_2004_3/Razdel9/03_Zadiraka.pdf (дата звернення 19.05.2025)

9. Стеганографія : навчальний посібник / О. О. Кузнецов, С. П. Євсєєв, О. Г. Король. — Х. : Вид. ХНЕУ, 2011. — 232 с. (Укр. мов.). URL: <https://repository.hneu.edu.ua/bitstream/123456789/2289/1/Стеганографія.pdf> (дата звернення 19.05.2025)

10. О. М. Олійник О. Ю. Софіна Розробка системи захисту зображень за допомогою цифрових водяних знаків з використанням частотних перетворень на python. — ВНТКП ВНТУ. Факультет інтелектуальних інформаційних технологій та автоматизації LIV Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2025). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23121/19655> (дата звернення 20.05.2025)

11. Твердохліб, Ю. В. Методи та інформаційна технологія комплексного оцінювання параметрів вейвлет–перетворення нестационарних сигналів : дис. ... канд. техн. наук : 05.13.06 / Твердохліб Юлія Володимирівна ; ХНЕУ. — 3., 2018. — 178 с. URL: <https://www.hneu.edu.ua/wp-content/uploads/2018/11/Tverdohleb-Yu.V.-Disertaciya-Vchena-rada-K-64.055.03-2018.pdf> (дата звернення 20.05.2025)

12. Дубровін, В. І. Огляд методів вейвлет-перетворення для аналізу фоно- та електрокардіограм / В. І. Дубровін, Ю. В. Твердохліб, А. В. Рашавченко // Журнал Біоніка інтелекту. — 2013. — № 2(8). — С. 87-92. URL:

<https://ela.kpi.ua/server/api/core/bitstreams/704ba9d2-eb96-4b5d-bafb-08bcaaef5716/content> (дата звернення 20.05.2025)

13. Lin, E.T., Video and Image Watermark Synchronization, in Center for Education and Research in Information Assurance and Security. PhD thesis, Purdue University: West Lafayette.2015. URL:

<https://engineering.purdue.edu/~ace/thesis/linet/lin-thesis.pdf> (дата звернення 20.05.2025)

14. Майданюк В.П. Навчальний посібник. Кодування зображень. – Вінниця: ВДТУ, 2000. – 62 с URL:

https://pdf.lib.vntu.edu.ua/books/2024/LAN/Maidanyuk_2009_164.pdf (дата звернення 20.05.2025)

15. Fisenko V. T. Komp'juternaja obrabotka i raspoznavanie izobrazhenij: ucheb. posobie / V. T. Fisenko, T. Ju. Fisenko. — SPb : SPbGU ITMO, 2008. — 192 p. URL: <https://oeipt.vntu.edu.ua/index.php/oeipt/article/view/420> (дата звернення 20.05.2025)

16. Основи програмування. Python. Частина 1 підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПП ім. Ігоря Сікорського – Київ : КПП ім. Ігоря Сікорського, 2018. – 195 с. URL:

<https://ela.kpi.ua/server/api/core/bitstreams/dbbe8ff5-11d7-4a92-918c-d1445c3d20a7/content> (дата звернення 21.05.2025)

17. Алгоритмічні мови та програмування. Частина 1. Основи інформатики та комп'ютерної техніки. Навчальний посібник. – Вінниця: ВНТУ, 2009. – 116с. URL:

https://pdf.lib.vntu.edu.ua/books/2024/LANZ/Dovhalets_P1_2009_116.pdf (дата звернення 21.05.2025)

18. D. J. MacKay: "Information Theory, Inference, and Learning Algorithms" // Cambridge University Press, 2003. - С. 354.

19. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / [L.-C. Chen, G. Papandreou, I.

Kokkinos, et al.] URL: <https://arxiv.org/pdf/1606.00915.pdf> (дата звернення 22.05.2025).

20. Матеріали XII науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 18–19 грудня 2024 р.). – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024. – 238 с. URL: https://elartu.tntu.edu.ua/bitstream/lib/29696/1/zbirnyk_2019_17_12_20_19.pdf (дата звернення 22.05.2025)

21. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с. URL: <http://stepanivka-school1.edukit.sumy.ua/Files/downloads/Book-Python.pdf> (дата звернення 23.05.2025)

22. Голубєв В.О., Гавловський В.Д., Цимбалюк В.С. Інформаційна безпека: проблеми боротьби зі злочинами у сфері використання комп'ютерних технологій. — Запоріжжя: Просвіта, 2001. — с.198-201. URL: <http://www.nbu.gov.ua/node/300> (дата звернення 24.05.2025)

23. Алгоритми та методи обчислень: навч. посіб. для студ. спеціальностей 121 «Інженерія програмного забезпечення», спеціалізації «Програмне забезпечення високопродуктивних комп'ютерних систем та мереж» та 123 «Комп'ютерна інженерія», спеціалізації «Комп'ютерні системи та мережі» / М. А. Новотарський; КПІ ім. Ігоря Сікорського – Київ : КПІ ім. Ігоря Сікорського, 2019. – 407 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/7421218e-d7dd-4e75-aa3e-bd7979db4e6d/content> (дата звернення 25.05.2025)

24. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York // John Wiley and Sons, 1996. URL: <https://mrjacse.wordpress.com/wp-content/uploads/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf> (дата звернення 25.05.2025)

25. Neil F. Johnson, Zoran Duric, Sushil Jajodia, Information Hiding : Steganography and Watermarking. - Attacks and Countermeasures. Kluwer Academic

Publishers. 2001. 160p. URL: <https://archive.org/details/informationhidin0000john> (дата звернення 25.05.2025)

26. G. Bouridane. A, M. K. Ibrahim. Digital Image Watermarking Using Balanced Multi wavelets / Bouridane G. – Transaction on Signal Processing. – IEEE. 2006, 1519-1536 pp. URL: https://www.researchgate.net/publication/3319616_Digital_image_watermarking_using_balanced_multiwavelets (дата звернення 23.05.2025)

27. OpenCV - Open Computer Vision Library. URL: <https://opencv.org> (дата звернення 17.05.2025)

ДОДАТКИ

Додаток А
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА СИСТЕМИ ЗАХИСТУ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ
ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ З ВИКОРИСТАННЯМ ЧАСТОТНИХ
ПЕРЕТВОРЕНЬ НА PYTHON

Алгоритм модуля роботи системи

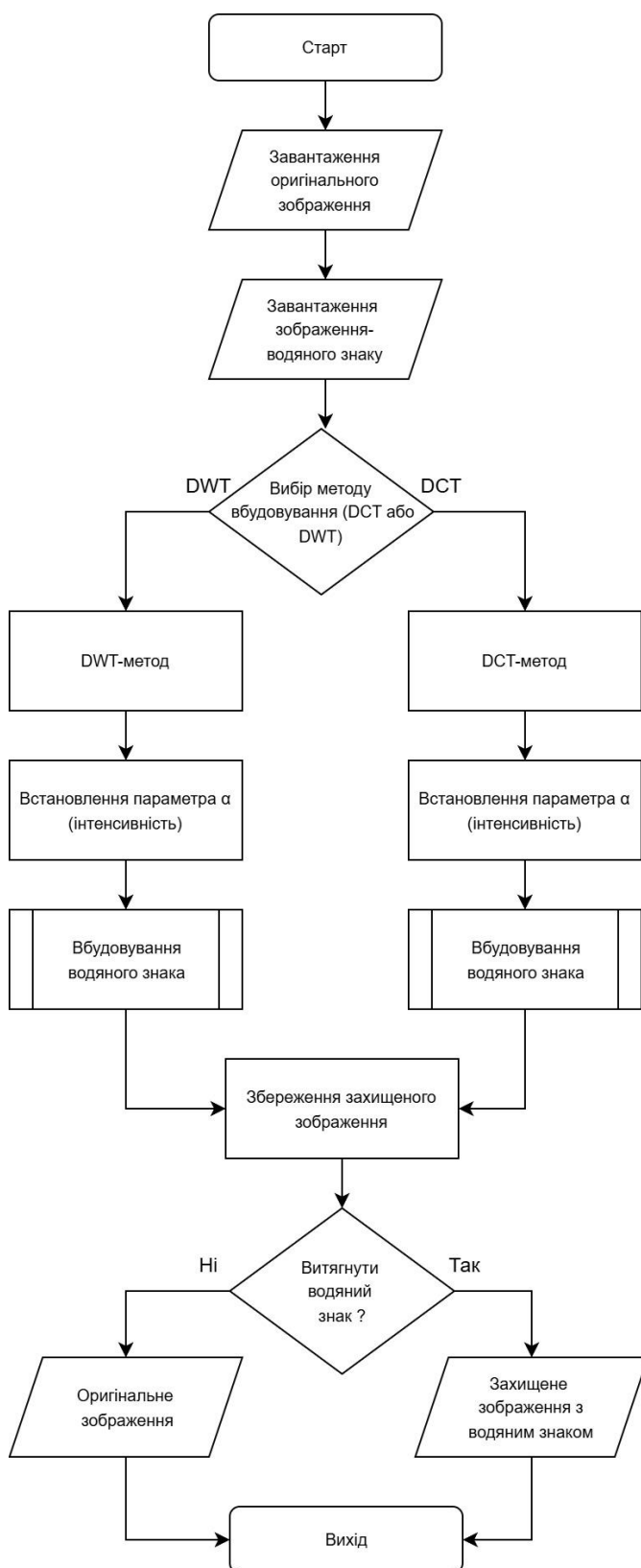


Рисунок А.1 – Алгоритм модуля роботи системи захисту зображень методами частотних перетворень

Алгоритм модуля тестування системи

Рисунок А.2 – Алгоритм модуля тестування системи захисту зображень

Схема інтеграції зі зовнішніми системами

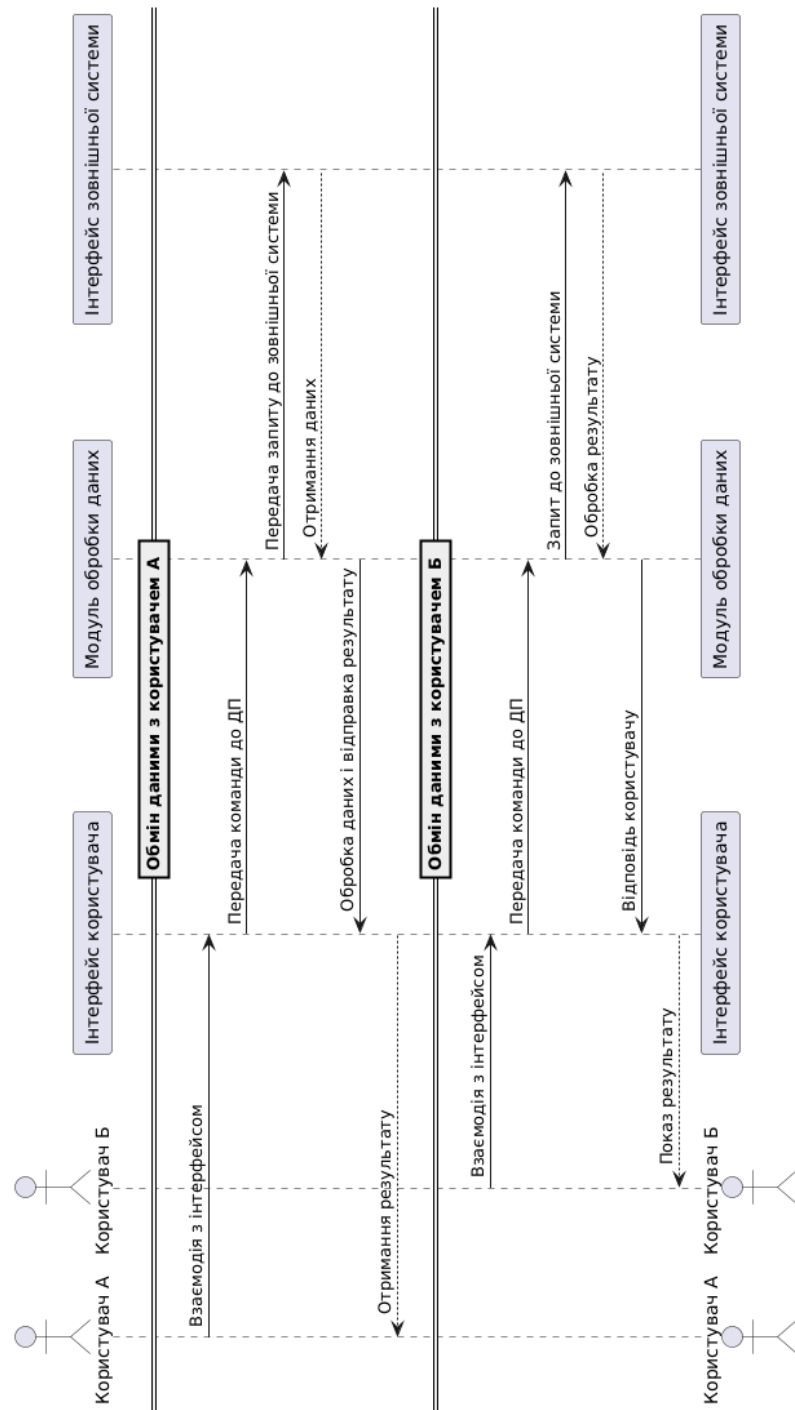


Рисунок А.3 – Схема інтеграції зі зовнішніми системами

Додаток Б
(обов'язковий)

Лістинг основних функцій програми

```
import os
import numpy as np
import cv2
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim
import pywt

class WatermarkSystem:
    def __init__(self, alpha=0.1, method="dct"):
        self.alpha = alpha
        self.method = method
        self.original_image = None
        self.watermark_image = None
        self.watermarked_image = None
        self.extracted_watermark = None

    def load_image(self, filepath, is_watermark=False):
        image = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
        if image is None:
            raise ValueError("Не вдалося завантажити зображення")
        if is_watermark:
            self.watermark_image = image
        else:
            self.original_image = image
        return image

    def save_image(self, image, filepath):
        ext = os.path.splitext(filepath)[1].lower()
```

```

if ext in ('.jpg', '.jpeg'):
    params = [cv2.IMWRITE_JPEG_QUALITY, 90]
    cv2.imwrite(filepath, image, params)
elif ext == '.png':
    params = [cv2.IMWRITE_PNG_COMPRESSION, 5]
    cv2.imwrite(filepath, image, params)
else:
    cv2.imwrite(filepath, image)

def embed_watermark(self):
    if self.original_image is None or self.watermark_image is None:
        raise ValueError("Завантажте оригінал та водяний знак")
    if self.method == "dct":
        self.dct_embed()
    elif self.method == "dwt":
        self.dwt_embed()
    else:
        raise ValueError("Невідомий метод вбудовування")

def dct_embed(self):
    watermark_resized = cv2.resize(self.watermark_image,
                                    (self.original_image.shape[1], self.original_image.shape[0]))
    original_dct = cv2.dct(np.float32(self.original_image) / 255.0)
    watermark_dct = cv2.dct(np.float32(watermark_resized) / 255.0)
    watermarked_dct = original_dct + self.alpha * watermark_dct
    self.watermarked_image = np.uint8(cv2.idct(watermarked_dct) * 255)

def dwt_embed(self):
    watermark_resized = cv2.resize(self.watermark_image,
                                    (self.original_image.shape[1] // 2, self.original_image.shape[0] // 2))
    coeffs_original = pywt.dwt2(self.original_image, 'haar')
    cA, (cH, cV, cD) = coeffs_original

```

```

coeffs_watermark = pywt.dwt2(watermark_resized, 'haar')
cA_w, (cH_w, cV_w, cD_w) = coeffs_watermark
cH_watermarked = cH + self.alpha * cH_w
coeffs_watermarked = (cA, (cH_watermarked, cV, cD))
self.watermarked_image = np.uint8(pywt.idwt2(coeffs_watermarked, 'haar'))

def extract_watermark(self):
    if self.original_image is None or self.watermarked_image is None:
        raise ValueError("Завантажте оригінал та вбудоване зображення")
    if self.method == "dct":
        self.dct_extract()
    elif self.method == "dwt":
        self.dwt_extract()
    else:
        raise ValueError("Невідомий метод витягування")

def dct_extract(self):
    original_dct = cv2.dct(np.float32(self.original_image) / 255.0)
    watermarked_dct = cv2.dct(np.float32(self.watermarked_image) / 255.0)
    extracted_dct = (watermarked_dct - original_dct) / self.alpha
    self.extracted_watermark = np.uint8(cv2.idct(extracted_dct) * 255)

def dwt_extract(self):
    coeffs_original = pywt.dwt2(self.original_image, 'haar')
    cA_o, (cH_o, cV_o, cD_o) = coeffs_original
    coeffs_watermarked = pywt.dwt2(self.watermarked_image, 'haar')
    cA_w, (cH_w, cV_w, cD_w) = coeffs_watermarked
    cH_extracted = (cH_w - cH_o) / self.alpha
    coeffs_extracted = (np.zeros_like(cH_extracted),
                        (cH_extracted, np.zeros_like(cH_extracted), np.zeros_like(cH_extracted)))
    self.extracted_watermark = np.uint8(pywt.idwt2(coeffs_extracted, 'haar'))
    if self.watermark_image is not None:

```

```
self.extracted_watermark = cv2.resize(self.extracted_watermark,
                                       (self.watermark_image.shape[1], self.watermark_image.shape[0]))
```

```
def calculate_metrics(self, img1, img2):
```

```
    if img1.shape != img2.shape:
        img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
    psnr_val = psnr(img1, img2)
    ssim_val = ssim(img1, img2)
    return psnr_val, ssim_val
```

```
def apply_attack(self, attack_type):
```

```
    if self.watermarked_image is None:
        raise ValueError("Спочатку вбудуйте водяний знак")
    attacked = self.watermarked_image.copy()
    if attack_type == "шум":
        noise = np.random.normal(0, 25, self.watermarked_image.shape)
        attacked = np.clip(attacked + noise, 0, 255).astype(np.uint8)
    elif attack_type == "обрізання":
        h, w = attacked.shape
        crop_h, crop_w = int(h * 0.1), int(w * 0.1)
        attacked = attacked[crop_h:-crop_h, crop_w:-crop_w]
        attacked = cv2.resize(attacked, (w, h))
    elif attack_type == "стиснення":
        encode_param = [int(cv2.IMWRITE_JPEG_QUALITY), 50]
        _, encimg = cv2.imencode('.jpg', attacked, encode_param)
        attacked = cv2.imdecode(encimg, cv2.IMREAD_GRAYSCALE)
    elif attack_type == "розмиття":
        attacked = cv2.GaussianBlur(attacked, (5, 5), 0)
    self.watermarked_image = attacked
```

```
# Приклад використання:
```

```
# ws = WatermarkSystem(alpha=0.1, method="dct")
```

```
# ws.load_image("original.png")
# ws.load_image("watermark.png", is_watermark=True)
# ws.embed_watermark()
# ws.save_image(ws.watermarked_image, "result.png")
# ws.apply_attack("mym")
# ws.extract_watermark()
# ws.save_image(ws.extracted_watermark, "extracted.png")
# psnr_val, ssim_val = ws.calculate_metrics(ws.original_image, ws.watermarked_image)
```

**Додаток В
(обов'язковий)
ПРОТОКОЛ**

**ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Розробка системи захисту зображень за допомогою цифрових водяних знаків з використанням частотних перетворень на Python

Тип роботи: бакалаврська кваліфікаційна робота

(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра АІТ, ФІТА, ІСТ-21Б

(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 9,19 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак 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 плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Олег БСІКАЛО, зав. каф. АІТ

(прізвище, ініціали, посада)

Людков ЯМЧУК, секретар каф. АІТ

(прізвище, ініціали, посада)

(підпис)

(підпис)

Особа, відповідальна за перевірку

(підпис)

Овчинников К.В.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

(підпис)

Софина О.Ю.

(прізвище, ініціали, посада)

Здобувач

(підпис)

Олійник О.М.

(прізвище, ініціали)