Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Лавренюк Олександри Василівни

на здобуття ступеня вищої освіти Бакалавра

Система ідентифікації джерела поширення графічних даних на основі частотного цифрового маркування

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ.220243.22.02.29 ПЗ

Виконала студентка 4 курсу група КБ-22-2 Олександра ЛАВРЕНЮК
Керівник д-р філософії Наталія ПЕТЛЯК
Нормоконтролер д-р філософії Наталія ПЕТЛЯК

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

3 06 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

9 січня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лавренюк Олександрі Василівні

1 Тема роботи Система ідентифікації джерела поширення графічних даних на основі частотного цифрового маркування

Керівник роботи доктор філософії Наталія ПЕТЛЯК

Затверджено наказом ректора університету від 8 січня 2026 р. № 7

2 Строк подання студентом кваліфікаційної роботи на кафедру 27 травня 2026 р.

3 Вихідні дані до роботи Проаналізувати наявні системи захисту електронного документообігу та методи ідентифікації інсайдерів. Дослідити існуючі методи цифрової стеганографії. Сформувати вимоги до системи ідентифікації джерела поширення графічних даних на основі частотного цифрового маркування: забезпечення невидимості маркування, стійкість до JPEG-стиснення, автоматизація генерації копій для співробітників. Розробити алгоритми вбудовування та виявлення міток на основі дискретного косинусного перетворення та квантування. Обґрунтувати вибір засобів розробки. Розробити прототип системи. Провести оцінку достовірності якості зображень та коректності зчитування міток після спотворень.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз систем захисту електронного документообігу та методів ідентифікації джерел витоку даних. Формування вимог до системи ідентифікації джерела поширення графічних даних. Розроблення алгоритмів частотного цифрового маркування на основі dct та qim. Теоретичні основи дискретного косинусного перетворення. Розроблення алгоритму вбудовування цифрової мітки. Розроблення алгоритму виявлення та зчитування мітки. Реалізація та оцінка достовірності прототипу системи. Обґрунтування вибору засобів розробки. Реалізація алгоритмів у програмному середовищі. Експериментальна перевірка достовірності системи.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Загальна структурна схема системи ідентифікації джерела витоку даних. Блок-схема алгоритму вбудовування цифрового водяного знаку. Блок-схема алгоритму вилучення та дешифрування мітки. Схема взаємодії модулів програмного комплексу (генерація, база даних, аналіз). Графіки залежності рівня помилок (BER) від ступеня стиснення зображення.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль			

7 Дата видачі завдання 12 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Формування вимог до системи ідентифікації джерела витоку графічних даних	Березень	
Розроблення алгоритмів частотного цифрового маркування на основі DCT та QIM	Квітень	
Розробка програмного прототипу системи ідентифікації джерела поширення графічних даних	Квітень	
Експериментальне дослідження працездатності прототипу та оцінка достовірності системи	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Травень	
Захист КР	Червень	

Студент Олександра ЛАВРЕНЮК

Керівник кваліфікаційної роботи Наталія ПЕТЛЯК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система ідентифікації джерела поширення графічних даних на основі частотного цифрового маркування.

Автор роботи: Лавренюк Олександра Василівна.

Керівник роботи: Петляк Наталія Сергіївна.

Пояснювальна записка: 77 с., 6 додатків, 18 рисунків, 1 таблиця, 55 джерел.

Графічна частина: 3 аркуші.

Ключові слова: витік даних, дискретне косинусне перетворення, ідентифікація інсайдера, квантування індексів, стеганографія, цифровий водяний знак.

Метою кваліфікаційної роботи є розробка прототипу системи ідентифікації джерела поширення конфіденційних графічних даних на основі частотного **цифрового маркування.**

У роботі розроблено алгоритм прихованого вбудовування ідентифікаційних даних у графічні документи. Як математичний апарат використано дискретне **косинусне перетворення у поєднанні з методом модуляції індексів квантування** та контентно-залежним підходом до вибору області маркування.

У результаті створено програмний прототип, який автоматизує процес **генерації персоналізованих копій та здійснює сліпе вилучення цифрового** водяного знака з перехоплених документів. Розроблене рішення може бути інтегроване в існуючі системи електронного документообігу для запобігання **інсайдерським витокам та встановлення особи порушника.**

25.05.2026

ABSTRACT

Subject of qualification work: System for identifying the source of graphic data distribution based on frequency digital watermarking.

Author: Lavreniuk Oleksandra Vasylivna.

Head of work: Petliak Nataliia Serhiivna.

Explanatory note: 77 p., 6 appendices, 18 figures, 1 table, 55 sources.

Graphic part: 3 sheets.

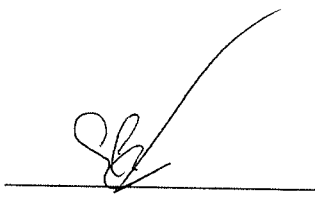
Keywords: data leakage, digital watermark, discrete cosine transform, insider identification, quantization index modulation, steganography.

The purpose of the qualification work is to develop a prototype of a system for identifying the source of confidential graphic data distribution based on frequency digital watermarking.

This paper presents an algorithm for the covert embedding of identification data into graphic documents. The mathematical framework employs the discrete cosine transform in combination with the quantization index modulation method and a content-dependent approach to selecting the embedding region.

As a result, a software prototype was created that automates the process of generating personalized copies and performs blind extraction of the digital watermark from intercepted documents. The developed solution can be integrated into existing electronic document management systems to prevent insider leaks and identify the violator.

25.05.2026



ЗМІСТ

ВСТУП

Стрімкий розвиток інформаційно-комунікаційних технологій зробив електронний документообіг невіддільною складовою функціонування підприємств та установ. Проте побудова надійного захисту корпоративних мереж від зовнішніх атак не вирішує проблеми витоку даних повністю. Найбільш непередбачуваною загрозою залишається внутрішній порушник – легітимний користувач системи, який має авторизований доступ до конфіденційної інформації. Сучасні системи контролю мережевого трафіку здатні ефективно блокувати несанкціоноване копіювання файлів, але вони виявляються безсилими перед проблемою «аналогового витоку», коли зловмисник фотографує екран або пересилає знімок через месенджери. У такому випадку файл втрачає оригінальні метадані, піддається стисненню та зашумленню, що унеможливлює встановлення джерела витоку традиційними методами.

Дієвим рішенням є застосування методів цифрової стеганографії – превентивного вбудовування невидимих ідентифікаційних маркерів у структуру графічного документа. Оскільки класичні просторові методи не витримують алгоритмічного стиснення з втратами, виникає гостра потреба у використанні методів частотної області. Поєднання дискретного косинусного перетворення з алгоритмами модуляції індексів квантування та адаптивним контентно-залежним вибором зон маркування здатне забезпечити оптимальний баланс між візуальною непомітністю та стійкістю до деструктивних впливів. З огляду на це, розробка системи ідентифікації джерела витоку на основі частотного маркування є вкрай актуальним науково-прикладним завданням.

Метою кваліфікаційної роботи є розробка прототипу системи ідентифікації джерела поширення конфіденційних графічних даних на основі частотного цифрового маркування, яка дозволить автоматизувати процес захисту документів та забезпечить високу ймовірність встановлення порушника у разі компрометації даних.

Об'єктом розроблення є процес захисту конфіденційних графічних даних від

несанкціонованого розповсюдження в інформаційно-комунікаційних системах.

Предметом кваліфікаційної роботи є математичні моделі, алгоритми та програмні засоби контентно-залежного вбудовування і надійного вилучення прихованих цифрових маркерів у частотній області зображень.

Для досягнення поставленої мети у роботі визначено низку завдань:

– формування комплексних вимог до системи маркування, що включають суворі критерії візуальної непомітності, стійкості до стиснення з втратами та можливості сліпого вилучення мітки без наявності еталонного файлу;

– дослідження та адаптація алгоритму прихованого вбудовування мітки, який базується на дискретному косинусному перетворенні, розрахунку локальної дисперсії для пошуку найскладнішої текстури та модифікації середньочастотних коефіцієнтів;

– розроблення надійного алгоритму сліпого вилучення прихованої інформації з інтеграцією лексичних фільтрів та механізмів мажоритарного голосування;

– програмна реалізація прототипу системи для забезпечення пакетної генерації персоналізованих документів;

– проведення комплексного експериментального дослідження розробленого рішення для оцінки його достовірності, візуальної якості маркованих зображень та визначення меж стійкості до цифрових спотворень.

1 АНАЛІЗ СИСТЕМ ЗАХИСТУ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ ТА МЕТОДІВ ІДЕНТИФІКАЦІЇ ДЖЕРЕЛ ВИТОКУ ДАНИХ

1.1 Аналіз існуючих методів ідентифікації інсайдерів

В сучасному бізнес-світі, однією з корінних загроз є не тільки проблема зловмисників і шахраїв, а й легітимних користувачів (внутрішніх порушників). Працівники організацій і користувачі з високим рівнем доступу несуть велику небезпеку для чутливих даних, тому що вже знайомі з внутрішніми процесами і знають, як діяти, щоб уникнути виявлення [1].

Саме через це антивірусів, спеціально побудованих демілітаризованих зон та мережевих екранів часто буває недостатньо для боротьби з витоками інформації. За звітом IBM про нанесену фінансову шкоду витоками даних за 2025 рік, середньою вартістю зловмисних атак зсередини приходиться близько 4,92 млн доларів США. Це вже другий рік поспіль, коли саме дії внутрішніх кадрів приносять багатомільйонні збитки [2]. В свою чергу, в звіті 2024 року було вказано, що саме цей тип атак призвів до витрат у розмірі 4,99 млн доларів США [3]. Графічне відображення можна побачити на рис.1.1, де значення вимірюються в мільйонах доларів США; відсоток від усіх випадків витоку даних.

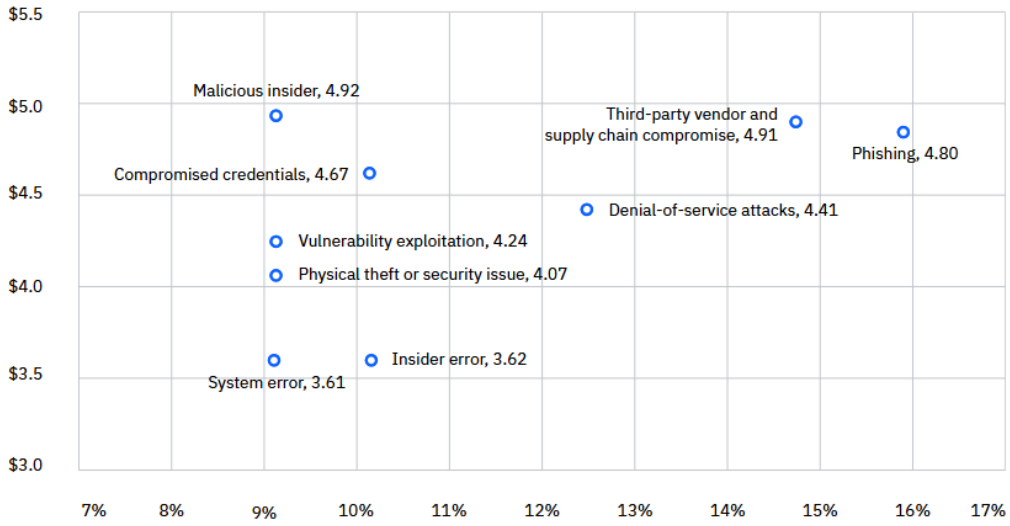


Рисунок 1.1 – Основні вектори атак та першопричини за даними звіту IBM за 2025 р. (мовою оригіналу) [2]

Етап авторизації хоч і захищає від несанкціонованого доступу, все ще не може вплинути на шкідливі дії внутрішніх порушників, або ж помилку користувача.

З традиційних методів виявлення внутрішніх загроз, часто застосовуються системи виявлення вторгнень (англ. Intrusion Detection Systems, IDS). Найпоширеніші це ті, що працюють методом виявлення вже відомих раніше загроз на основі їх сигнатур. Цей підхід часто є ефективним, проте безсилий проти раніше невідомих шкідливих програм або кодів, яких ще немає в базах даних. А поведінкові IDS, працюючі методами виявлення аномалій, в свою чергу, можуть подолати цей недолік. Даний тип систем аналізує дії працівників і користувачів, вимірюючи, наскільки вони відходять від припустимої норми.

Проте внутрішня загроза не може бути усунена на 100% і в цьому випадку, так як якщо користувач знає про існування такої системи в організації, він може цілеспрямовано змінювати свою поведінку, щоб уникнути виявлення. Це призведе до збільшення кількості помилкових спрацювань [4].

Для більш ефективного захисту корпоративних даних багато сучасних організацій впроваджують системи системи запобігання витоку даних (англ. Data Loss Prevention, DLP). Основна задача таких систем полягає у контролі конфіденційної інформації у трьох станах: під час зберігання, передачі, а також використання на робочих станціях. DLP-системи можуть з легкістю блокувати спробу скопіювати конфіденційний документ на зовнішній USB-накопичувач, заборонити його відправлення через особисту електронну пошту і навіть заблокувати функцію створення зображення екрана стандартними системними засобами [5].

Проте, незважаючи на ефективність у роботі з цифровим контентом, класичні DLP-системи мають вразливість, так звану проблему «аналогового витоку» (англ. analog hole) [6]. Оскільки такий процес повністю обходить мережеві протоколи та файлову систему комп’ютера, DLP-система не розпізнає порушень встановлених політик безпеки, що робить організацію вразливою до витоку.

Щоб компенсувати недоліки класичних IDS та DLP-систем, індустрія інформаційної безпеки перейшла до застосування технологій поведінкової аналітики користувачів та сутностей (англ. User and Entity Behavior Analytics, UEBA). На відміну від базових систем виявлення аномалій, UEBA використовує складні алгоритми машинного навчання для створення профілю «нормальної» поведінки як людей, так і мережевих пристроїв, серверів, тощо [7]. Так, якщо працівник, який зазвичай відкриває кілька текстових документів щодня, раптово починає завантажувати великі об’єми графічних файлів у значних кількостях, або звертається до певної бази даних у не типовий для роботи час, система UEBA зафіксує аномалію, розрахує показник ризику (risk score) та надішле попередження адміністраторам [8].

Системи UEBA не є стовідсотковим методом для ідентифікації джерел витоку. Вони працюють з розрахунками ймовірностей, проте не надають жодного прозорого 100% доказу. Якщо фото конфіденційного креслення чи документа було розміщене, наприклад, у закритому Telegram-каналі, інформація від UEBA може лише допомогти звузити коло підозрюваних осіб, які мали доступ до документа в даний проміжок часу. Вони не зможуть юридично підтвердити, хто з цих людей є поширювачем.

Отож, що превентивні методи контролю трафіку та аналізу поведінки є обов’язковими, проте не достатніми для повноцінного захисту. Для гарантованого виявлення внутрішнього порушника у випадку «аналогового» витоку єдиним ефективним способом є попереднє розміщення прихованих ідентифікаторів у самих зображеннях за допомогою методів цифрової стеганографії.

1.2 Дослідження методів цифрової стеганографії і водяних знаків

У випадку захисту конфіденційної графічної інформації від інсайдерських витоків, звичайні методи шифрування виявляються безсилими. Після отримання і розшифрування документа легітимним користувачем інформація стає вразливою

до фізичного і цифрового копіювання (наприклад, фотографування екрана монітора або друку, розповсюдження файла в месенджерах). Для вирішення цієї проблеми застосовують методи цифрової стеганографії, зокрема технологію цифрових водяних знаків (ЦВЗ), яка дозволяє непомітно вписати ідентифікаційну інформацію безпосередньо у графічний контейнер [9].

ЦВЗ можна розділити за різними характеристиками: за видимістю (видимі та приховані), за стійкістю (крихкі, напівстійкі та робастні) та за методом вилучення (сліпі та несліпі) [10]. Для систем ідентифікації джерела витоку найбільше підходять приховані, робастні і сліпі ЦВЗ. Прихованість гарантує, що зловмисник не помітить наявності мітки; робастність забезпечує виживання мітки після деструктивних перетворень (стиснення, кадрування); а сліпе вилучення дозволяє виявити мітку без наявності оригінального (немаркованого) зображення, що є обов'язковою умовою при аналізі перехоплених витоків даних [11].

З математичної точки зору вбудовування ЦВЗ, всі відомі методи цифрової стеганографії поділяються на дві великі категорії: методи просторової області (Spatial Domain) та методи області перетворень чи частотної області (Frequency/Transform Domain) [12].

1.2.1 Методи просторової області

Методи просторової області безпосередньо оперують значеннями пікселів (їх яскравістю або кольоровими компонентами RGB). Найпопулярнішим і найпростішим алгоритмом у цій групі є метод найменш значущого біта (англ. Least Significant Bit, LSB). [13].

Ідея методу LSB полягає у заміні останніх (молодших) бітів байтового представлення пікселя на біти секретного повідомлення . Оскільки зміна найменш значущого біта змінює колір пікселя всього на 1 градацію (з 256 можливих для 8-бітного каналу), людське око не в змозі зафіксувати різницю.

Коли АЦП (аналого-цифровий перетворювач) перетворює напругу в цифрову форму, один цифровий код складається з діапазону напруги. Цей діапазон напруги, що відповідає одному коду, називається LSB, або найменш значущим бітом. LSB – це найменший рівень, який може перетворити АЦП

(найменший приріст напруги, який може виявити АЦП), або найменший приріст, який може видати ЦАП (цифро-аналоговий перетворювач) [14].

АЦП потребує опорної напруги для перетворення аналогового сигналу в цифровий. Залежно від кількості бітів, він ділить опорну напругу на невеликі значення. Нижче на рисунку 1.2 зображено приклад: якщо це 8-бітний АЦП, значення будуть виглядати як: 1, 2, 3, … і так до 256.

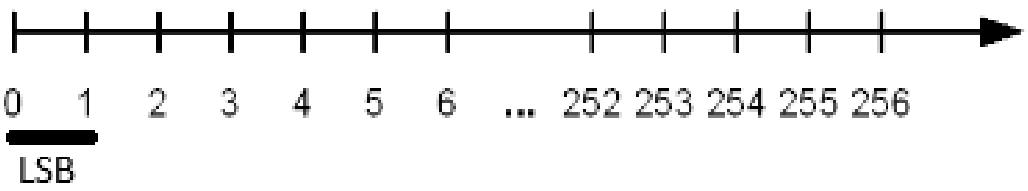


Рисунок 1.2 – Приклад розрахунків

Одна одиниця виміру дорівнює 1 LSB і визначається як: $LSB = \frac{\vee ref}{N^2}$, де N – кількість бітів [11].

Переваги просторових методів:

– швидке виконання та низька обчислювальна складність;

– великий об'єм інформації (можна вбудувати великий обсяг даних).

Недоліки просторових методів:

– вразливість при стисненні. Стискання за допомогою JPEG із втратою інформації остаточно знищує найменш значущі біти, а разом з ними – і водяний знак;

– низька стійкість до геометричних атак (масштабування, поворот) та зашумлення (наприклад, при фотографуванні екрана на камеру смартфона);

– ризик виявлення стегоаналітикою [15].

З огляду на ці недоліки, застосування просторових методів (LSB) в системах DLP (Data Loss Prevention) абсолютно неефективне, оскільки інсайдерський витік зазвичай супроводжується саме перетворенням формату або захопленням екрана.

1.2.2 Методи частотної області

Для того, щоб ЦВЗ були робастними, застосовуються техніки, що змінюють

зображення з просторової області у частотну. Замість того, щоб змінювати пікселі, алгоритми змінюють спектральні коефіцієнти зображення. Основні методи цього класу – це дискретне косинусне перетворення (англ. Discrete Cosine Transform, DCT), дискретне хвильове перетворення (англ. Discrete Wavelet Transform, DWT) та дискретне перетворення Фур'є (англ. Discrete Fourier Transform, DFT).

DFT – це метод, що дозволяє розглянути зображення у вигляді комплексної частотної площини. Вбудовування у амплітудний спектр DFT робить ЦВЗ інваріантним (стійким) до циклічного зсуву та масштабування [16]. Проте, цей метод вимагає складних операцій з комплексними числами і є чутливим до локальних обрізок зображення.

Також іншим методом є DWT. При застосуванні DWT зображення розбивається на частотні піддіапазони: один низькочастотний (LL) та три високочастотні (HL, LH, HH), які відповідають за вертикальні, горизонтальні та діагональні деталі [17]. Вбудовування мітки зазвичай відбувається у середньо- або високо-частотні смуги для збереження якості, або у низькочастотну смугу (LL) для максимальної робастності. DWT є дуже потужним інструментом, але він вимагає значних обчислювальних ресурсів і часто надійного вилучення мітки можливе лише за наявності оригінального зображення.

Наступним методом є DCT. DCT є найпопулярнішим методом у сучасній обробці зображень, оскільки саме це перетворення використовується у форматі стиснення JPEG. При застосуванні DCT зображення розбивається на блоки розміром 8×8 пікселів. Кожен блок перетворюється у матрицю частотних коефіцієнтів, де у верхньому лівому куті зосереджені низькі частоти (основа зображення), а у нижньому правому – високі (дрібні деталі та шум) [18-23].

Для вбудовування цифрових водяних знаків найчастіше обирають середні частоти. Зміна низьких частот призведе до спотворень (появи артефактів), а вбудовування у високі частоти є безглуздим, бо саме ці частоти алгоритми стиснення JPEG відкидають (обнуляють) для зменшення розміру файлу.

Слід особливо виділити метод вбудовування інформації у DCT-коефієнти за

допомогою алгоритму квантування індексів (англ. Quantization Index Modulation, QIM). Квантована індексна модуляція – алгоритм вбудовування водяних знаків, який модулює дані носія вбудованою інформацією, набув популярності завдяки своїй здатності протидіяти перешкодам користувача та доведеній ефективності з точки зору стійкості до спотворення [24-26].

Коли йдеться про збереження растрових зображень, основні компоненти – це частотні коефіцієнти, які ми отримуємо після поділу зображення на 8x8 піксельних блоків та застосування DCT-перетворення. ЦВЗ поділяються на дві категорії: значущі та незначущі. На відміну від підходів, що використовують беззмістовні псевдовипадкові послідовності, для підвищення надійності доцільно застосовувати значущий водяний знак. Ним виступає структурований текстовий ідентифікатор співробітника, який конвертується у бінарну послідовність, що складається з інформаційних бітів 0 та 1.

У типових системах алгоритм QIM модулює певний середньочастотний коефіцієнт. Вибір частот в роботі обумовлений властивостями людської візуальної системи (HVS): модифікація нижчих частот спричиняє появу помітних "блокових" артефактів, тоді як вищі частоти безповоротно обнуляються алгоритмами компресії JPEG. Коефіцієнт DCT забезпечує оптимальний баланс, оскільки зміни в ньому сприймаються оком як природний високочастотний шум, але при цьому він зберігається після проходження через фільтри соціальних мереж. Правило квантування можна описати таким чином: початкове значення коефіцієнта ділиться на заздалегідь визначений розмір кроку квантування Q (наприклад, оптимальним експериментальним значенням вважається Q=62) та округлюється до найближчого цілого числа (індексу). Алгоритм тоді досліджує парність індексу, яку він отримує, якщо решта від ділення індексу на 2 відповідає цільовим бітом повідомлення (0 або 1). Якщо залишок збігається, то значення залишається основним. А в протилежному випадку, індекс буде зкориговано на +-1 у бік найменшої математичної похибки, тобто, зсувається до найближчого правильного парного або непарного числа. На фінальному етапі відбувається множення квантового значення на крок Q, після чого блок буде піддаватись

зворотному перетворенню (IDCT).

Цей метод дозволяє досягти унікального балансу: він забезпечує "сліпе" вилучення (не потрібен оригінал), високу швидкість роботи та дуже високу стійкість до JPEG-компресії і фотографування.

1.2.3 Порівняльний аналіз та обґрунтування вибору методу

Для проектування ефективної системи ідентифікації джерела витоку конфіденційних графічних даних необхідно обрати підхід, який задовольняє всім ключовим вимогам: стійкість до стиснення, збереження візуальної якості та здатність до сліпого детектування. Зведену порівняльну характеристику розглянутих методів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз методів вбудовування цифрових водяних знаків

Характеристика / Метод	LSB (Просторова)	DWT (Хвильова)	DFT (Фур'є)	DCT + QIM (Косинусна)
Стійкість до JPEG-стиснення	Низька (руйнується)	Висока	Середня	Дуже висока
Стійкість до шуму (фото/скан)	Низька	Висока	Висока	Висока
Вплив на візуальну якість	Непомітний	Непомітний	Помітний	Непомітний
Сліпе вилучення (Blind)	Так	Частково (залежить від алгоритму)	Ні	Так (повністю сліпе)
Обчислювальна складність	Низька	Дуже висока	Висока	Середня (оптимальна)

Як випливає з таблиці вище і результатів порівняльного аналізу [27, 28], методи просторової області (наприклад LSB) виявляються абсолютно непридатними для задачі ідентифікації джерела витоку інформації через їх надзвичайну вразливості до спотворень. Будь-яке перезбереження файлу остаточно руйнує прихований ідентифікатор. В той же час більш стійкі частотні методи (DWT, DFT) характеризуються високою робастністю, проте є надмірно

обчислювально складними і в більшості випадків вимагають наявності оригінального (немаркованого) зображення при експертизі, що ускладнює розслідування інцидентів [29]. Слід також зауважити, що хоча останні розробки методів стеганографії і водяних знаків на основі генеративно-змагальних нейронних мереж (GAN) відзначаються дуже високою робастністю маркування [30], їх застосування у вбудованих у DLP системи корпоративних сценаріях для генерації сотень документів в реальному часі є невиправдано дорогим і ресурсозатратним методом у порівнянні з класичними математичними перетвореннями.

Враховуючи вищезазначене, найбільш адекватним для поставленої задачі (ідентифікація внутрішнього порушника за фото документа з месенджера) було обрано метод дискретного косинусного перетворення у поєднанні з алгоритмом квантування індексів. Застосування багатовимірного QIM дає можливість добитися високої точності "сліпого" вилучення даних, забезпечуючи при цьому надійний захист вбудованої інформації без істотної деградації візуальної якості.

Вбудування мітки в середньочастотні коефіцієнти DCT робить водяний знак стійким до агресивного стиснення алгоритмами соціальних мереж та месенджерів. Більш того, обраний підхід дає теоретичну можливість зберегти цілісність ідентифікатора навіть при геометричних перетвореннях, таких як кадрування (вирізання фрагмента) або масштабування [31], які є неминучими при фотографуванні екрана монітора смартфоном або створенні часткових знімків документа.

Саме це математичне рішення, що дає можливість безпомилково прочитати ідентифікатор співробітника (ID) із «злитого» файлу, буде використано у розробленому програмному модулі маркування у наступних розділах.

Одним із найважливіших вдосконалень класичного частотного маркування є перехід від глобального (загального) до адаптивного контентно-залежного вбудовування (Content-Aware Watermarking). Звичайні методи на основі DCT здійснюють стеганографічне маркування по всій площині зображення, через що на однотонних (гладких) ділянках картинки може з'являтися помітний

структурний шум, що псує візуальне сприйняття [32]. Як показало дослідження людської візуальної системи (англ. Human Visual System, HVS), людське око дуже чутливе до зміни пікселів на чистому фоні, проте майже неможливо помітити такі ж за інтенсивністю артефакти, якщо вони заховані у складних текстурах та/або контурах зображення [33-36].

Щоб розв’язання цієї задачі та зменшення зони втручання в оригінальний файл, у системі застосовується метод сегментування зображення на області та маскування текстур (Variance-based Texture Masking). Алгоритм віртуально ділить зображення на фіксовану сітку розміром 4×4 (16 незалежних зон). Замість того, щоб виконувати маркування "наосліп", система заздалегідь розраховує дисперсію яскравості пікселів кожної області [37].

Дисперсія виступає об'єктивним критерієм текстурної складності: порожні ділянки (білі зони) матимуть дисперсію, близьку до нуля, а місця з великим скупченням тексту, таблиць, штампів або логотипів будуть мати високу мінливість значень пікселів. Для вбудовування інформації вибирається та область, що має найбільшу дисперсію. Вона функціонує як HVS-маска, що повністю поглинає візуальні зміни від процесу квантування DCT-коефіцієнтів, роблячи артефакти невидимими для людського ока, зловмисника чи навіть систем стеганоаналізу [38-39].

Цей підхід дозволяє розташувати ЦВЗ на площі, що не перевищує 6-10% від загальної площі документа. Незважаючи на зменшення простору для вбудовування, оптимізація обсягу корисного навантаження (використання лише ПІБ та ID співробітника) дозволяє багато разів повторити цю інформацію в межах однієї текстурної області (забезпечити локальну надмірність). Під час процесу сліпого вилучення декодер паралельно сканує всі сегменти зображення: порожні області генерують фільтрований "білий шум", а цільова обрана макро-зона віддає чітку послідовність маркерів, яка відновлюється за допомогою методу мажоритарного голосування [40-42]. Комбінація DCT, QIM та контентно-залежного вибору зони маркування гарантує, що знак буде "виживати" навіть після дуже агресивного стиснення алгоритмами JPEG (із показником якості нижче

70%), що є типовим для сучасних соціальних мереж і месенджерів [43-45].

1.3 Формування вимог до системи ідентифікації джерела витоку графічних даних

Вивчення недоліків традиційних систем DLP і вразливостей у просторовій області до стеганографії дозволяє чітко визначити, якими властивостями має володіти розроблена програмна система. Сьогодні однією з ключових проблем електронного документообігу є ситуація, коли легальний користувач (інсайдер) незаконно розповсюджує за межі організації або підприємства конфіденційний документ у вигляді графічного зображення (наприклад, через месенджери, які агресивно стискають картинку, обрізаючи важливу інформацію документа) або зберігає файл на зовнішній накопичувач без збереження оригінальних метаданих.

Теоретичне проєктування такої системи спирається на розв'язання фундаментального протиріччя цифрової стеганографії, відомого як «стеганографічний трикутник». Його суть полягає у взаємозалежності трьох параметрів: обсягу вбудованих даних (ємності), візуальної якості (непомітності) та стійкості до перетворень (робастності). Оскільки неможливо одночасно максимізувати всі три показники, при формуванні вимог необхідно визначити пріоритетність характеристик відповідно до специфіки корпоративного документообігу. Для завдань ідентифікації джерела витоку пріоритет зазвичай надається робастності та непомітності, тоді як ємність обмежується мінімально необхідним набором ідентифікаторів.

Для протидії таким загрозам і отримання результату, що дозволяє точно встановити особу порушника в разі інциденту, перспективна система частотного цифрового маркування повинна відповідати набору жорстких функціональних і технічних вимог:

1. Вимоги до непомітності інформації (Imperceptibility).

Однією з ключових характеристик вбудованого цифрового водяного знака є

відсутність його помітності для візуального сприйняття людини. Інформація, вкладена у зображення, не має привертати увагу потенційних інсайдерів, адже навіть найдрібніші візуальні ефекти (артефакти, розмиття, зниження чи підвищення контрастності тощо) можуть стати причиною застосування зловмисником додаткових фільтрів, спрямованих на знищення знака [46]. У науковому середовищі широко використовується параметр PSNR (Peak Signal-to-Noise Ratio) як критерій непомітності. Так, у роботах, присвячених цифровій стеганографії, зазначається, що реалізована система має забезпечувати PSNR не менше 35-40 дБ для збереження візуальної ідентичності маркованого зображення та оригіналу [47].

2. Вимоги до робастності (Robustness) та захисту від атак.

Тим, хто поширює зображення, майже ніколи не вдається поширити його в оригінальному вигляді. Зображення піддається, незалежно від нашого бажання, різним деструктивним операціям. Найпоширенішими з них є стиснення з втратами (наприклад, стиснення, яке застосовується месенджерами, такими як Telegram чи Viber, стиснення у соціальних мережах, які конвертують файли у формат JPEG зі зниженням якості для економії трафіку тощо) та кадрування для приховання логотипів чи номерів документів [48]. Таким чином, маркування має бути робастним, як до випадкових модифікацій, так і до спрямованих атак, що мають на меті знищення прихованих даних. Саме через таку вимогу розробники змушені відмовлятися від чутливих до атак просторових методів (зокрема LSB) на користь методів у частотній області [49-52]. Застосування DCT в комплексі з QIM дає змогу вбудувати мітку в середньочастотний діапазон, який зазнає деструкції при стисненні у форматі JPEG найменше [53].

3. Вимога до ідентифікації та інформаційної ємності.

Щоб ЦВЗ міг служити доказом у суді чи в адміністративних процедурах, він повинен містити достатньо інформації для однозначної ідентифікації джерела витоку (конкретної особи). Ємність алгоритму має дозволяти приховано вбудовувати ідентифікаційний мета-рядок, але обов’язково бути оптимізованою для забезпечення максимальної просторової надмірності (Spatial Redundancy). До

системи висувається вимога здатності закодувати та зберегти такий мінімально необхідний набір конкретних для кожного співробітника ідентифікаторів:

– унікальний табельний номер або ID співробітника;

– прізвище та ініціали співробітника.

Зменшення обсягу навантаження дозволяє реалізувати механізм мажоритарного відновлення даних. Це теоретично обґрунтована вимога: якщо інформаційний рядок короткий, його можна вбудувати в документ десятки разів. Під час зчитування це дає змогу відновити оригінальне повідомлення навіть за умови руйнування 50-70% фрагментів маркування, що критично важливо при аналізі скриншотів чи знімків частин документа.

В ході розробки було прийнято відмовитись від вбудовування надлишкових метаданих (назва відділу, службові примітки). Зменшення обсягу інформаційного навантаження дозволяє збільшити кількість циклів дублювання водяного знака у межах доступної площі зображення. Саме такий підхід гарантує виживання мітки та її успішне відновлення після агресивного стиснення алгоритмами сучасних месенджерів (Telegram, Viber).

4. Вимога до сліпого вилучення

Ключовою вимогою до розробки системи є можливість декодера (модуля аналізу) вилучати приховане повідомлення без необхідності мати доступ до оригіналу (чистого) зображення [54]. У реальних випадках розслідування, працівник відділу безпеки має у своєму розпорядженні лише перехоплений файл (часто змінений, стиснений тощо). Алгоритм читання (експертизи) має базуватися тільки на аналізі частотних коефіцієнтів самого перехопленого файлу і не має потребувати порівняння з еталонним зразком. Метод QIM найкращим чином відповідає цій вимозі, оскільки вилучення бітів здійснюється за допомогою обчислення відстані до найближчого кроку квантування [55].

5. Вимоги до архітектури та автоматизації.

Враховуючи потреби сучасного корпоративного та державного документообігу, описана система не має бути набором різних скриптів. Програмний комплекс повинен мати модульну архітектуру і надавати

максимальний рівень автоматизації маркування документів під час їх розсилання працівникам. Серед основних архітектурних вимог можна відзначити:

– наявність модуля для інтеграції з реляційною базою даних (наприклад, MS SQL Server) для динамічного імпорту актуального списку співробітників та їхніх ідентифікаторів;

– можливість масової (пакетної) генерації маркованих копій: система повинна вміти за один цикл обчислень вивести десятки унікальних персоналізованих копій одного документа для різних отримувачів;

– наявність інтуїтивно зрозумілого графічного інтерфейсу користувача (GUI), який логічно розділяє процеси на модуль генерації (для спеціалістів з документообігу) та модуль аналізу/експертизи.

Також, важливою технічною вимогою є забезпечення мінімального рівня ймовірності хибнопозитивного спрацювання (False Positive Rate). Система не повинна знаходити «фантомні» мітки там, де їх немає, оскільки помилкове звинувачення співробітника може призвести до серйозних репутаційних та юридичних наслідків для організації.

Крім суто технічних та алгоритмічних характеристик, при проєктуванні подібних систем слід обов'язково враховувати й організаційно-психологічний аспект їхнього впровадження. Наявність інтегрованої системи цифрового маркування в корпоративному середовищі виконує не лише функцію технічного розслідування інцидентів, але й відіграє потужну превентивну (стримуючу) роль.

Якщо співробітники організації офіційно поінформовані про те, що кожен конфіденційний документ, який відображається на їхньому робочому моніторі, містить їхній персональний невидимий ідентифікатор, ймовірність навмисного витоку інформації знизиться. Потенційний інсайдер усвідомлює, що будь-який зроблений ним знімок екрана або переслaний файл може бути пов'язаний з його обліковим записом. Це повністю нівелює хибне відчуття безкарності та технічної анонімності, яке зазвичай і спонукає порушників до ігнорування політик інформаційної безпеки. Таким чином, впровадження вимог невидимого маркування переводить концепцію захисту від реактивної моделі (пошук винного

після того, як дані вже опинилися у відкритому доступі) до проактивної, коли самому факту витоку запобігають ще на рівні психології користувача.

Виконання цих вимог допоможе розробити комплексне рішення, яке буде усувати вразливі місця традиційних засобів захисту і надасть надійний інструмент для превентивного маркування і розслідування інсайдерських витоків конфіденційних графічних даних.

1.4 Постановка задачі

На основі проведеного аналізу можна зробити висновок, що проблема витоків конфіденційної інформації та захисту внутрішніх ресурсів залишається надзвичайно актуальною. Технології DLP та UEBA демонструють високу ефективність у контролі мережевої активності, проте виявляються вразливими до використання обхідних каналів передачі інформації (наприклад, зовнішніх накопичувачів або мобільних месенджерів). У таких випадках превентивне стеганографічне маркування, яке забезпечує надійність доказової бази та можливість точної ідентифікації джерела витоку (інсайдера), є одним із найбільш дієвих методів захисту.

Аналіз сучасних методів вбудовування даних показав, що просторові методи (зокрема LSB) не забезпечують необхідного рівня стійкості до деструктивних впливів, таких як стиснення з втратами (JPEG), що є типовим при передачі зображень через месенджери та соціальні мережі. Натомість використання частотної області, зокрема DCT у комбінації з QIM та контентно-залежним вибором зон маркування, забезпечує оптимальний баланс між візуальною непомітністю та робастністю до зовнішніх атак.

З огляду на вищезазначене, метою кваліфікаційної роботи є розробка прототипу системи ідентифікації джерела поширення конфіденційних графічних даних на основі частотного цифрового маркування, яка дозволить автоматизувати процес захисту документів та забезпечить високу ймовірність встановлення

порушника у разі компрометації даних.

У роботі необхідно вирішити такі основні завдання:

– провести аналіз предметної області, визначити вразливості існуючих систем електронного документообігу та обґрунтувати необхідність застосування стеганографічних методів для протидії інсайдерським витокам;

– сформувати комплексні вимоги до системи цифрового маркування, що включають критерії візуальної непомітності (об'єктивна метрика PSNR не менше 35 дБ), стійкості до алгоритмів стиснення JPEG та забезпечення можливості сліпого вилучення водяного знаку без наявності оригінального файлу;

– дослідити та адаптувати алгоритм вбудовування цифрової мітки, що базується на розкладанні зображення на блоки 8x8 пікселів, застосуванні прямого DCT, розрахунку локальної дисперсії для контентно-залежного маскування та модифікації середньочастотних коефіцієнтів методом QIM;

– розробити алгоритм вилучення та декодування прихованої інформації, що включає механізми мажоритарного голосування для фільтрації похибок та розпізнавання оптимізованого структурованого ідентифікатора (ПІБ співробітника та унікальний табельний номер);

– здійснити програмну реалізацію прототипу системи (DLP-модуля маркування) з використанням мови програмування Python та бібліотеки комп'ютерного зору OpenCV, забезпечивши інтеграцію з реляційною базою даних для автоматизації генерації персоналізованих копій документів;

– провести експериментальне дослідження розробленого програмного комплексу для оцінки достовірності системи, перевірки якості маркованих зображень та визначення меж стійкості алгоритму до типових цифрових спотворень.

Розв'язання поставлених завдань дозволить створити дієвий програмний інструмент, який може бути інтегрований у корпоративні чи державні системи документообігу для підвищення рівня конфіденційності та забезпечення невідворотності відповідальності за витік інформації.

2 РОЗРОБЛЕННЯ АЛГОРИТМІВ ЧАСТОТНОГО ЦИФРОВОГО МАРКУВАННЯ НА ОСНОВІ DCT ТА QIM

2.1 Теоретичні основи дискретного косинусного перетворення (DCT)

Дискретне косинусне перетворення є одним із ключових математичних інструментів у сфері цифрової обробки зображень та стеганографії. Воно дає змогу перетворити сигнал із просторової області, де дані представлені у вигляді значень яскравості пікселів, у частотну область, де зображення описується у вигляді амплітуд частотних гармонік. Основною перевагою DCT порівняно з іншими спектральними перетвореннями є його властивість ущільнення енергії (energy compaction): більшість візуально значущих інформаційних значень припадають на небагату кількість низькочастотних коефіцієнтів, що дає змогу ефективно відділяти важливі структурні деталі від високочастотного шуму та надлишкових даних.

У стеганографічних алгоритмах та стандартах компресії зображень (наприклад, JPEG), зображення зазвичай розбивається на незалежні матричні блоки розміром 8x8 пікселів, і для кожного блоку застосовується двовимірне пряме дискретне косинусне перетворення (2D-DCT). Формула прямого перетворення для матриці яскравостей f(x, y) розміром NxN (де для алгоритму маркування N=8) наведена у вигляді:

$$(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\left[\frac{(2x+1)u\pi}{16}\right]cos\left[\frac{(2y+1)v\pi}{16}\right], \quad (2.1)$$

де F(u,v) – отримане значення частотного коефіцієнта (амплітуда просторової частоти);

u – індекс просторової частоти по горизонталі (u = 0, 1, ..., 7);

v – індекс просторової частоти по вертикалі (v = 0, 1, ..., 7);

$\frac{1}{4}$ – масштабувальний коефіцієнт перетворення, що розраховується як $\frac{2}{N}$ для блоку розміром N=8;

C(u) – нормалізувальний множник по горизонталі, який дорівнює $\frac{1}{\sqrt{2}}$ при u = 0, та 1 при u > 0;

C(v) – нормалізувальний множник по вертикалі, який дорівнює $\frac{1}{\sqrt{2}}$ при v = 0, та 1 при v > 0;

Σ – математичний оператор підсумовування (знак сигма);

x – просторова координата пікселя по горизонталі всередині блоку (x = 0, 1, ..., 7);

y – просторова координата пікселя по вертикалі всередині блоку (y = 0, 1, ..., 7);

f(x,y) – вихідне значення яскравості пікселя за координатами x та y у просторовому домені до перетворення;

π – математична константа (число пі);

16 – константа знаменника, що визначається базовою формулою як 2N, де розмір блоку N=8.

Для відновлення зображення після вшивання стеганографічного маркування у частотній області застосовують зворотне двовимірне дискретне косинусне перетворення (2D-IDCT). Ця операція перетворює змінену матрицю коефіцієнтів F'(u, v) назад у просторову область записуючи нові значення яскравості пікселів f'(x, y) для подальшого збереження зображення. Формула зворотного перетворення має наступний вигляд:

$$f'(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F'(u,v)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]. \quad (2.2)$$

де f'(x,y) – відновлене (реконструйоване) значення яскравості пікселя за просторовими координатами x та y після застосування зворотного перетворення;

$\frac{1}{4}$ – масштабувальний коефіцієнт зворотного перетворення (визначається як $\frac{2}{N}$ для блоку розміром N=8);

Σ – математичний оператор підсумовування (знак сигма);

u – індекс просторової частоти по горизонталі (u = 0, 1, ..., 7);

v – індекс просторової частоти по вертикалі (v = 0, 1, ..., 7);

C(u) – нормалізувальний множник по горизонталі, який дорівнює $\frac{1}{\sqrt{2}}$ при u = 0, та 1 при u > 0;

C(v) – нормалізувальний множник по вертикалі, який дорівнює $\frac{1}{\sqrt{2}}$ при v = 0, та 1 при v > 0;

F'(u,v) – квантоване (модифіковане в процесі вбудовування цифрової мітки) значення частотного коефіцієнта за координатами u та v;

x – просторова координата реконструйованого пікселя по горизонталі всередині блоку (x = 0, 1, ..., 7);

y – просторова координата реконструйованого пікселя по вертикалі всередині блоку (y = 0, 1, ..., 7);

π – математична константа (число пі);

16 – константа знаменника, що визначається базовою формулою перетворення як 2N, де N=8.

Оскільки базові функції косинуса є ортогональними, перетворення повністю зворотне із мінімальними обчислювальними похибками. Відповідно, зображення після зворотного перетворення у просторовий домен повністю ідентичне маркованому, що забезпечує умову непомітності цифрового водяного знака.

Саме застосування зворотного двовимірного дискретного косинусного перетворення є обов'язковим кроком у частотній стеганографії. Після того, як алгоритм замінює деякі частотні коефіцієнти, формуючи нову матрицю F'(u, v), ці дані залишаються у частотному просторі, у якому їх не можна безпосередньо побачити звичайним графічним засобом. Функцією 2D-IDCT є роль математичного мосту, що дозволяє побачити частотні амплітуди у вигляді матриці пікселів. З точки зору фізики, зворотне перетворення знаходить кожне значення пікселя f'(x, y) як суму всіх 64 двовимірних косинусних базисних

функцій для блоку, де ваговими множниками виступають модифіковані коефіцієнти F'(u, v) .

Важливою технічною особливістю при застосуванні 2D-IDCT у цифрових системах є перетворення з дійсних чисел у цілочисельний формат машинного подання кольору. Результатом обчислення зворотного перетворення f'(x, y) є число з плаваючою комою (floating-point number). Однак для збереження зображення у стандартних 8-бітних кольорових просторах (наприклад, YCrCb або RGB), значення яскравості мають бути цілими числами в строго обмеженому діапазоні. Саме тому після виконання основного IDCT всі пікселі зазвичай проходять через операції математичного округлення (rounding) та відсікання (clipping) значень, які виходять за межі діапазону. Математично підсумкове значення реконструйованого пікселя можна описати таким чином:

$$f'_{final}(x,y) = max\left(0, min\left(255, round\left(f'^{(x,y)}\right)\right)\right), \quad (2.3)$$

де $f'_{final}(x,y)$ – фінальне цілочисельне значення яскравості реконструйованого пікселя за просторовими координатами x та y, яке готове для збереження у файл графічного формату;

max – математична функція знаходження максимального значення з двох аргументів, що використовується для відсікання від'ємних похибок;

0 – нижня межа допустимого діапазону яскравості (мінімально можливе значення для 8-бітного формату, що відповідає абсолютно чорному кольору);

min – математична функція знаходження мінімального значення з двох аргументів, що використовується для відсікання значень, які перевищують ліміт;

255 – верхня межа допустимого діапазону яскравості (максимально можливе значення для 8-бітного формату, що відповідає абсолютно білому кольору);

round – математична функція стандартного округлення дійсного числа з плаваючою комою до найближчого цілого значення;

f'(x,y) – проміжне реконструйоване значення яскравості пікселя за

координатами x та y після застосування зворотного перетворення (IDCT), яке через похибки обчислень може виходити за межі діапазону [0, 255] та мати дробову частину;

x – просторова координата пікселя по горизонталі всередині блоку (x = 0, 1, ..., 7);

y – просторова координата пікселя по вертикалі всередині блоку (y = 0, 1, ..., 7).

Округлення і відсікання при просторовій реконструкції призводять до появи неминучої обчислювальної похибки (truncation error), яка викликає нелінійні спотворення піксельних значень. Дана похибка є першим руйнівним фактором, що впливає на вбудований ЦВЗ на шляху збереження і стиснення файлу.

Через наявність похибки IDCT-реконструкції категорично необхідно застосовувати стеганографічні методи з великим запасом стійкості. Саме через це в розробленому в роботі алгоритмі застосовується метод QIM з широким кроком квантування (наприклад, Q=65). Такий крок формує достатньо широкий "захисний інтервал" для зміненого коефіцієнта, що гарантує, що зворотні просторові спотворення після 2D-IDCT не приведуть до помилкової ідентифікації інформаційного біта під час подальшого прямого перетворення 2D-DCT в процесі детекції.

Після застосування прямого двовимірного дискретного косинусного перетворення (2D-DCT) до блоку зображення розміром 8×8 пікселів відбувається кардинальна зміна способу подання візуальної інформації. У просторовій області зображення задається масивом значень яскравості окремих пікселів, тоді як після DCT цей самий блок можна розглядати як суперпозицію 64 ортогональних двовимірних косинусних базисних функцій із різними просторовими частотами. Кожний коефіцієнт перетворення F(u, v) визначає вагову частку відповідної функції базису у відновленні зображення.Однією з найважливіших властивостей DCT є ефект ущільнення енергії (energy compaction), який полягає в тому, що для більшості природних і текстурних зображень більша частка спектральної енергії

сконцентрована у відносно невеликій кількості низькочастотних коефіцієнтів. Це пов’язано з високою просторовою кореляцією сусідніх пікселів: яскравість фону зазвичай змінюється повільно, а різкі контури займають порівняно невелику площу. Через це низькочастотні базисні функції несуть основний зміст або загальну структуру зображення, тоді як високочастотні передусім описують локальні межі, дрібні текстури або навіть візуальний шум.

У результаті перетворення формується матриця з 64 коефіцієнтів, серед яких лівий верхній елемент F(0, 0) є постійною складовою (DC-коефіцієнт, від англ. direct current), а решта 63 коефіцієнти розглядаються як змінні складові (AC-коефіцієнти, від англ. alternating current). Постійна складова має особливий фізичний зміст, оскільки її значення пропорційне середньому значенню яскравості всього матричного блоку 8×8 :

$$F(0,0) \propto \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y)\,, \tag{2.4}$$

де F(0,0) – коефіцієнт постійної складової (DC-коефіцієнт), що відповідає нульовим просторовим частотам (u=0, v=0) і визначає середню інтенсивність (яскравість) всього блоку;

∝– математичний символ пропорційності, що вказує на лінійну залежність між значенням коефіцієнта та сумою значень пікселів;

Σ – математичний оператор підсумовування (знак сигма);

x – просторова координата пікселя по горизонталі в межах обраного блоку (x = 0, 1, ..., 7);

y – просторова координата пікселя по вертикалі в межах обраного блоку (y = 0, 1, ..., 7);

f(x,y) – значення яскравості (амплітуди) пікселя у просторовій області за відповідними координатами x та y.

Внаслідок цього DC-коефіцієнт зазвичай має найбільшу абсолютну

величину і сконцентровує більшість спектральної енергії. Змінні складові, навпаки, описують коливання яскравості: AC-коефіцієнти з малими значеннями індексів u та v відповідають повільним, плавним змінам яскравості, а коефіцієнти з великими індексами фіксують швидкі, різкі зміни пікселів. Наприклад, коефіцієнт F(1, 0) описує лише горизонтальні зміни яскравості, F(0, 1) – вертикальні, а F(7, 7) вказує на наявність високочастотного шуму в обох напрямках одночасно.

Математична властивість ущільнення енергії, за якої абсолютна більшість інформації фокусується у лівому верхньому куті просторово-частотної матриці, наочно продемонстрована на тривимірній гістограмі (рис. 2.1).

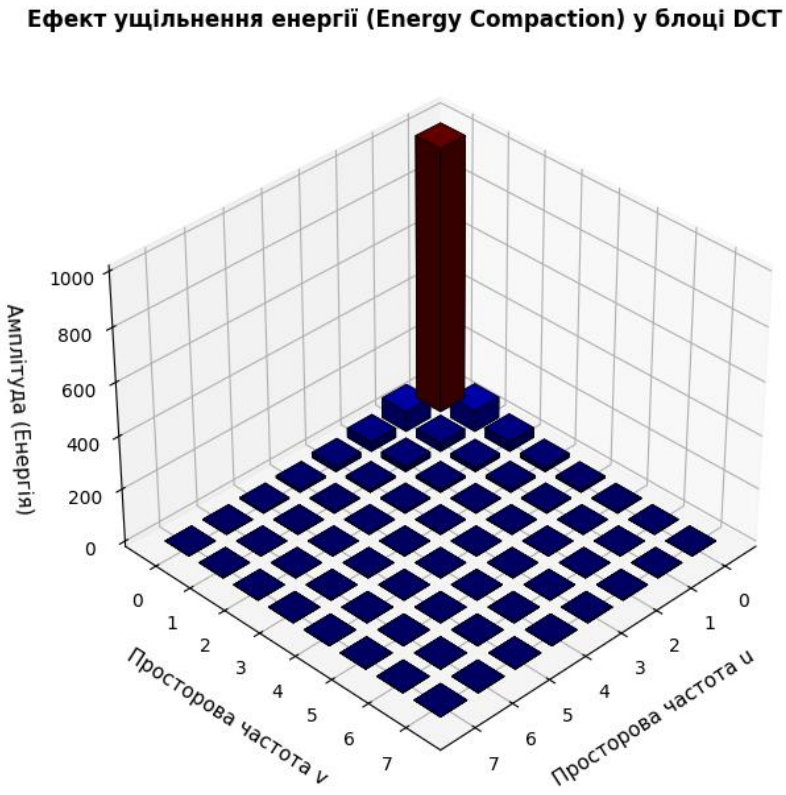


Рисунок 2.1 – Тривимірна візуалізація розподілу спектральної енергії в матриці DCT

Для цілей стиснення зображень (зокрема алгоритмів, використаних у стандарті JPEG) та цифрової стеганографії матрицю частот зазвичай умовно

розбивають на низькочастотну (англ. low frequency, LF), середньочастотну (англ. mid frequency, MF) та високочастотні області (англ. high frequency, HF). Низькочастотна область охоплює DC-компоненту та кілька найближчих сусідніх коефіцієнтів, які описують загальну структуру зображення. Високочастотна область знаходиться в нижньому правому куті матриці і відповідає деталям. Середньочастотна область описує основні текстурні характеристики та локальні переходи.

Оскільки частоти тут зростають по діагоналі, цей поділ здійснюється за допомогою зигзагоподібної адресації матриці (англ. zig-zag scan). Принципове значення цього напряму сканування полягає в концентрації найбільш інформативних коефіцієнтів на початку послідовності, що безпосередньо впливає на вибір позицій для вбудовування цифрових водяних знаків. Розподіл частотних областей у матричному блоці розміром 8x8 пікселів, напрямок зигзагоподібного сканування та просторове розташування цільового середньочастотного коефіцієнта F(3,2) наочно проілюстровано на рисунку 2.2.

Рисунок 2.2 – Структура DCT-матриці та зигзагоподібне сканування з виділенням цільового коефіцієнта

Визначення оптимальної частоти для реалізації вбудованої маркувальної інформації є складним інженерним компромісом між трьома взаємопов'язаними ознаками: непомітністю (англ. imperceptibility), стійкістю (англ. robustness) та ємністю корисного навантаження (англ. capacity). Модифікація низькочастотних коефіцієнтів забезпечує велику стійкість інформації до стиснення, але призводить до значного погіршення візуальної якості. Оскільки кожен такий коефіцієнт впливає на велику область при зворотному перетворенні, це може викликати блокові артефакти, спотворення одноколірних або градієнтних областей та зміну локальної яскравості, що суперечить вимозі непомітності.

З іншого боку, вбудовування у високочастотні коефіцієнти гарантує ідеальне маскування, але такі зміни є абсолютно нестійкими. Алгоритми стиснення з втратами (зокрема JPEG та фільтри соціальних мереж) використовують матриці квантування, які цілеспрямовано обнуляють високі частоти для зменшення розміру файлу, що призводить до втрати прихованої інформації.

Враховуючи всі ці фактори, середньочастотна область є найбільш відповідним місцем для реалізації стеганографічного маркування. Зміни середніх частот практично непомітні для людського ока, особливо на текстурних ділянках, і водночас ці коефіцієнти не піддаються різкому обнуленню при стисненні та зберігаються при численних рекомпресіях. Для методу маркування на основі квантування індексів, який застосовується у проєктованій системі, цей аспект є вирішальним. Успішність QIM-детектування залежить від збереження співвідношень амплітуд між коефіцієнтами. Якщо маркер розміщено у середніх частотах, то навіть після спотворень від стиснення, шумової фільтрації та обчислювальних помилок при зворотному IDCT, залишається достатній запас між зонами квантування (англ. intervals of decision). Це гарантує, що величина коливань коефіцієнта не перевищуватиме половини кроку квантування $\frac{Q}{2}$, мінімізуючи ймовірність хибного зчитування цільового біта і формуючи надійний механізм ідентифікації джерела інформації.

На стійкість стеганографічного маркування мають вирішальний вплив алгоритми компресії з втратами, зокрема стандарт JPEG, який широко використовується у сучасних месенджерах та соціальних мережах. Ці алгоритми працюють на основі відсікання візуально надлишкової інформації, і саме високочастотні (HF) коефіцієнти матриці DCT найбільше зазнають найбільших спотворень у цьому процесі. Під час стиснення такі коефіцієнти діляться на відповідні значення із таблиць квантування яскравості, внаслідок чого вони переважно обнуляються. Відповідно, будь-яке маркування, вбудоване у високочастотний діапазон, буде повністю втрачено навіть при незначному стисненні графічного файлу.

З іншого боку, масштабні втручання у низькочастотний (LF) діапазон є небажаними через особливості людської візуальної системи (HVS). Оскільки низькі частоти, зібрані навколо DC-коефіцієнта, формують основний градієнт та середню яскравість блоку 8×8, то будь-яка їхня штучна зміна викликає структурні спотворення. На практиці це проявляється у вигляді ефекту "блочності" (blocking artifacts), коли межі змінених блоків стають контрастними і видимими на однотонних ділянках зображення, тим самим розкриваючи наявність стеганографічної мітки.

Враховуючи ці жорсткі математичні та візуальні обмеження, оптимальним підходом є використання середньочастотного діапазону (MF), який є компромісом між непомітністю і стійкістю до деструктивних факторів. У представленому алгоритмі маркування цільовим носієм інформації обрано конкретний середньочастотний коефіцієнт з координатами u=3 та v=2, тобто F(3, 2). Згідно з матрицею зигзагоподібного сканування (zig-zag scan), цей коефіцієнт знаходиться у вигідній транзитній зоні: його частота достатньо висока, щоб зміни, внесені алгоритмом QIM, сприймалися оком як незначний текстурний шум, але водночас досить низька, щоб уникнути агресивного обнулення при компресії JPEG.

Використання QIM саме для коефіцієнта F(3, 2) дозволяє інформаційним бітам зберегти цілісність навіть після того, як документ буде оброблений месенджерами. Правильний вибір координат мінімізує загальне погіршення якості

документа і дає достатній запас стійкості для надійного сліпого вилучення маркування під час розслідувань.

Ключовим питанням при застосуванні частотних методів цифрового маркування є вибір колірного простору для вбудовування даних. Зазвичай цифрові зображення представлені у колірному просторі RGB (або BGR, що є форматом за замовчуванням у бібліотеці комп'ютерного зору OpenCV), де кожен піксель формується поєднанням червоного, зеленого та синього кольорів. Проте ці колірні канали в RGB мають високий ступінь кореляції. Тому приховане повідомлення, вбудоване безпосередньо в один із цих каналів, неминуче викличе локальне порушення колірного балансу, що зробить водяний знак помітним. Щоб уникнути цього, перед застосуванням дискретного косинусного перетворення зображення конвертується в інший колірний простір.

У розробленому модулі для цього застосовується простір YCrCb. У моделі YCrCb візуальна інформація розділяється на три складові: яскравість (Y), яка визначає загальну чорно-білу структуру (ступінь освітленості) зображення, та дві колірно-різницеві складові (Cr і Cb), що відповідають за забарвлення (червоно-різницевий та синьо-різницевий канали відповідно). Завдяки такому підходу можна повністю відокремити структурні деталі та контури документа від його кольору.

Наочну візуалізацію процесу декомпозиції документа у колірному просторі YCrCb наведено на рис. 2.3. Як видно з наведеного зображення, складова яскравості (Y) зберігає абсолютно всі структурні контури, основний текст та загальну освітленість. Водночас хроматичні канали (Cr та Cb) мають вигляд майже суцільного нейтрально-сірого тла, на якому фіксуються лише незначні сліди кольорових елементів (наприклад, синього підпису чи герба). Це практично підтверджує той факт, що алгоритми компресії (зокрема, субдискретизація кольору в JPEG), які для економії розміру файлу цілеспрямовано відкидають інформацію з каналів Cr та Cb, не зруйнують стеганографічну мітку, якщо вона буде вбудована виключно в канал яскравості.

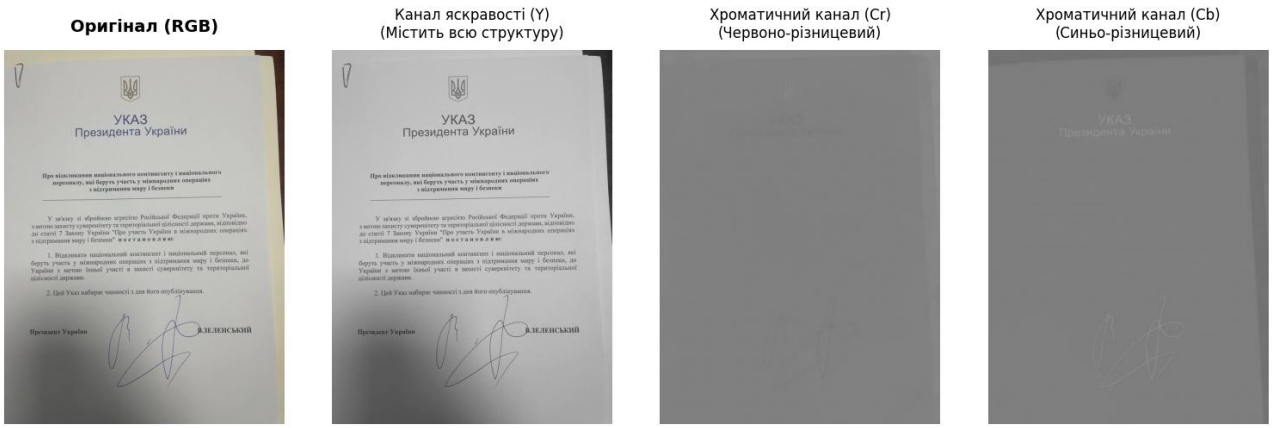


Рисунок 2.3 – Розкладання візуальної інформації документа на канали колірного простору YCrCb

Канал яскравості (Y) було обрано для вбудовування частотних маркерів з двох основних причин: через біологічні особливості людського зору (HVS) та принципи стиснення даних в алгоритмах сімейства JPEG. З фізіологічної точки зору, людське око має значно більше рецепторів для сприйняття яскравості та контрасту, ніж для розпізнавання кольорів. Відповідно, алгоритми стиснення з втратами (які широко використовуються у месенджерах та соціальних мережах) зазвичай застосовують субдискретизацію кольору (chroma subsampling). Під час стиснення вони цілеспрямовано відкидають значну частину інформації з каналів Cr та Cb, тоді як матриця каналу Y зберігається максимально наближеною до оригіналу. Тому вбудовування цифрового водяного знака у колірні канали неминуче призведе до його руйнування при будь-якому перезбереженні зображення.

Отже, робота з каналом Y є єдиним надійним і стійким до спотворень варіантом для відстеження витоків. У запропонованому програмному рішенні під час маркування зображення розкладається на три канали, після чого у матриці яскравості Y формується сітка з блоків 8x8 пікселів, у які і вбудовується ідентифікатор за допомогою методів DCT та QIM. Після модифікації коефіцієнтів середніх частот і виконання зворотного перетворення (IDCT), оновлений канал яскравості об'єднується з оригінальними колірними каналами (Cr та Cb), і зображення конвертується у вихідний формат для збереження. Завдяки цьому

ЦВЗ, що ідентифікує співробітника, зберігається навіть за умови агресивної компресії, а дані порушника можуть бути надійно вилучені під час розслідування інциденту.

2.2 Розроблення алгоритму вбудовування цифрової мітки

Процес приховування ідентифікаційних даних у графічний документ є складною багатокроковою процедурою, що поєднує попередню підготовку інформаційного навантаження, адаптивний аналіз текстурних властивостей зображення-контейнера, спектральне вбудовування у частотній області та просторову реконструкцію маркованого зображення. Звичайні методи шифрування виявляються безсилими у випадку захисту конфіденційної графічної інформації від інсайдерських витоків, оскільки після легітимного розшифрування документа інформація стає вразливою до фізичного копіювання. Для систем ідентифікації джерела витоку найбільше підходять приховані, робастні і сліпі ЦВЗ. Візуальне подання послідовності кроків розробленого алгоритму розділено на дві частини для кращого сприйняття: етап підготовки даних та пошуку цільової зони наведено у Додатку А, а основний цикл частотного вбудовування та просторової реконструкції зображення – у Додатку Б. Наукова новизна запропонованого алгоритму полягає у поєднанні адаптивного контентно-залежного вибору області вбудовування за критерієм максимальної дисперсії, просторового дублювання бітового навантаження та модифікації середньочастотних коефіцієнтів дискретного косинусного перетворення за допомогою методу квантування індексів. На відміну від підходів, що використовують беззмістовні псевдовипадкові послідовності, для підвищення надійності доцільно застосовувати значущий водяний знак. Комбінований підхід забезпечує суттєве підвищення робастності до зовнішніх атак без помітного зниження візуальної якості документа, усуваючи недоліки просторових методів, які є абсолютно неефективними в системах DLP через їх руйнування при

стисненні.

Логіку роботи алгоритму розпочинає етап формування інформаційного навантаження. На вхід подається структурований текстовий рядок, що містить ідентифікатор користувача (ім'я) та його службовий код: $M = Name|Code$. Оскільки при сліпому вилученні (blind watermarking) декодеру необхідно точно знати межі корисного навантаження для відсікання випадкового шуму, до повідомлення додається службовий маркер завершення (у програмній реалізації обрано комбінацію ###), утворюючи рядок $M' = M + \backslash text\{"\#\#\#"\}$. Далі текст перетворюється у бітову послідовність $B = \{b_1, b_2, \dots, b_{n\}}$, де $b_i \in \{0, 1\}$, а кожен символ кодується 8-бітним бінарним поданням. Особливістю алгоритму є використання просторової надмірності (spatial redundancy): після оцінки загальної ємності обраної макро-зони у блоках $C = N_{blocks}$, обчислюється кількість можливих повторів повідомлення $R = \left\lfloor \frac{C}{n} \right\rfloor$. Зменшення обсягу інформаційного навантаження дозволяє збільшити кількість циклів дублювання водяного знака у межах доступної площі зображення. Згенерований бітовий потік дублюється, утворюючи послідовність $B_R = B^R$, що повністю заповнює доступну область. Фактично це відіграє роль примітивного помилкостійкого кодування, яке гарантує виживання мітки та її успішне відновлення після агресивного стиснення алгоритмами сучасних месенджерів за допомогою методу мажоритарного голосування.

Схематичний принцип заповнення доступної спектральної ємності шляхом просторового дублювання інформаційних бітів наведено на рисунку 2.5.

Паралельно з обробкою тексту алгоритм виконує підготовку оригінального графічного контейнера. Зчитування зображення у вигляді матриці пікселів I(x,y) реалізовано через побітове завантаження масиву байтів за допомогою numpy.fromfile з подальшим декодуванням структури функцією cv2.imdecode. Це забезпечує коректну підтримку кириличних символів у шляхах файлової системи Windows. Оскільки вбудовування інформації безпосередньо у колірний простір BGR є неефективним і призводить до її руйнування при компресії, зображення

конвертується у яскравісно-колірний простір YCrCb та розкладається на незалежні компоненти I = (Y, Cr, Cb). Для подальших маніпуляцій використовується виключно канал яскравості $Y \in R^{h \times w}$, так як саме він зберігає найбільшу енергетичну стабільність після JPEG-компресії. З метою уникнення похибок округлення під час обчислень матриця Y примусово переводиться у формат чисел з плаваючою комою (float32).

Рисунок 2.5 – Схема формування просторової надмірності цільового бітового потоку

Для мінімізації візуальних спотворень документа застосовується контентно-залежний підхід до вибору області вбудовування. Звичайні методи на основі DCT здійснюють стеганографічне маркування по всій площині зображення, через що на однотонних ділянках картинки може з'являтися помітний структурний шум, що псує візуальне сприйняття. Матриця Y віртуально розбивається на 16 незалежних макро-зон (стандартна сітка 4x4). Кожна зона Z_k оцінюється за дисперсією

значень пікселів:

$$Var(Z_k) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2, \tag{2.5}$$

де $Var(Z_k)$ – значення дисперсії (міра текстурної складності та розкиду яскравості) для k-ї макро-зони зображення;

Z_k – k-та цільова макро-зона (сегмент) зображення, що підлягає аналізу;

$\frac{1}{N}$ – нормалізувальний множник, що визначається як обернене значення до загальної кількості пікселів у сегменті;

i – порядковий індекс поточного пікселя в межах досліджуваної зони (i = 1, 2, ..., N);

N – загальна кількість пікселів у поточному сегменті;

x_i – значення яскравості (інтенсивності) і-го окремого пікселя всередині зони;

μ – середнє арифметичне значення яскравості всіх пікселів у поточній зоні Z_k;.

Алгоритм циклічно аналізує всі сегменти та обирає цільову зону $Z^* = \arg max_k Var(Z_k)$, тобто ділянку з максимальною текстурною складністю. Цей механізм цілком спирається на властивості людської візуальної системи: людське око дуже чутливе до зміни пікселів на чистому фоні, проте майже неможливо помітити такі ж за інтенсивністю артефакти, якщо вони заховані у складних текстурах та/або контурах зображення. Саме це явище, відоме як perceptual masking, лежить в основі запропонованого критерію вибору макро-зони за максимальною дисперсією. Фактично алгоритм виконує адаптивне енергетичне маскування цифрової мітки, вбудовуючи її в ті місця, де природний вміст контейнера статистично приховує внесені модифікації. Завдяки чому забезпечується високе значення imperceptibility навіть при використанні порівняно великого кроку квантування Q. Цей підхід дозволяє розташувати ЦВЗ

на площі, що не перевищує 6-10% від загальної площі документа. Схематичне подання процесу адаптивного аналізу документа та вибору цільової макро-зони для маскування цифрової мітки наведено на рисунку 2.5.

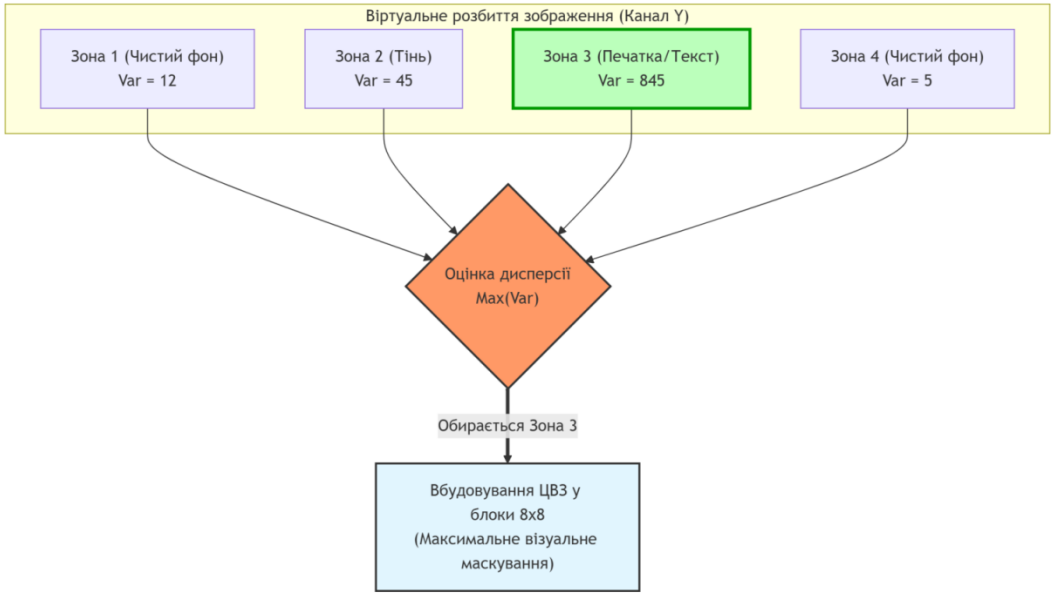


Рисунок 2.5 – Принцип контентно-залежного вибору цільової зони вбудовування на основі локальної дисперсії

Безпосереднє вбудовування інформації здійснюється у частотній області цільової зони Z*, яка, у свою чергу, розбивається на незалежні блоки розміром 8x8 пікселів. До кожного блоку B_{ij} застосовується операція прямого дискретного косинусного перетворення, утворюючи матрицю просторових частот $D_{ij} = DCT(B_{ij})$. З отриманого спектра вилучається цільовий середньочастотний коефіцієнт $c = D_{ij}(3,2)$. Зміна низьких частот може призвести до структурних спотворень (появи візуальних артефактів), а вбудовування у високі частоти є недоцільним, оскільки саме ці частоти алгоритми стиснення JPEG алгоритмічно відкидають (обнуляють) для зменшення розміру файлу. Вибір саме цієї координати обґрунтований оптимальним компромісом між непомітністю мітки та її стійкістю до деградації високочастотних складових.

Ключовою відмінністю реалізованого підходу від класичних адитивних схем (англ. additive watermarking) є застосування модуляції QIM, де інформація

кодується не амплітудою зміщення коефіцієнта, а його належністю до певного класу квантування. Квантована індексна модуляція набула популярності завдяки своїй здатності протидіяти перешкодам користувача та доведеній ефективності з точки зору стійкості до спотворення. Задаються дві області прийняття рішення (decision regions), визначені парністю:

$$Q_0 = \{2kQ\}, Q_1 = \{(2k+1)Q\}, \quad (2.6)$$

де Q_0 – множина значень (реконструкційних точок) квантування, що призначена для вбудовування інформаційного біта «0»;

Q_1 – множина значень (реконструкційних точок) квантування, що призначена для вбудовування інформаційного біта «1»;

k – довільне ціле число ($k \in \mathbb{Z}$), яке виступає множником та визначає порядковий індекс поточного інтервалу квантування;

$\mathbb{Z}$ – загальноприйняте математичне позначення множини всіх цілих чисел;

Q – обраний фіксований крок квантування, що визначає ширину інтервалу та відстань між сусідніми точками (згідно з налаштуваннями розробленої системи Q = 65);

$2k$ – парний множник, який гарантує математичну належність модифікованого коефіцієнта до множини Q_0;

$2k+1$ – непарний множник, який гарантує математичну належність модифікованого коефіцієнта до множини Q_1.

На практиці обчислюється квантований індекс $q = round\left(\frac{c}{Q}\right)$. Біт b_i вбудовується через перевірку парності: якщо $q\,(mod\,2) = b_i$, значення залишається без змін. Якщо умова не виконується, індекс коригується на одиницю ($q = q \pm 1$), причому напрям зміщення обирається так, щоб мінімізувати локальну математичну похибку та спектральне спотворення, зсуваючись до найближчого правильного парного або непарного числа. Оновлений коефіцієнт $c' = q \times Q$ замінює оригінальне значення у матриці

частот. Графічну інтерпретацію процесу модуляції індексів квантування та логіку мінімізації спектральної похибки під час вбудовування інформаційних бітів «0» та «1» наведено на рисунку 2.6.

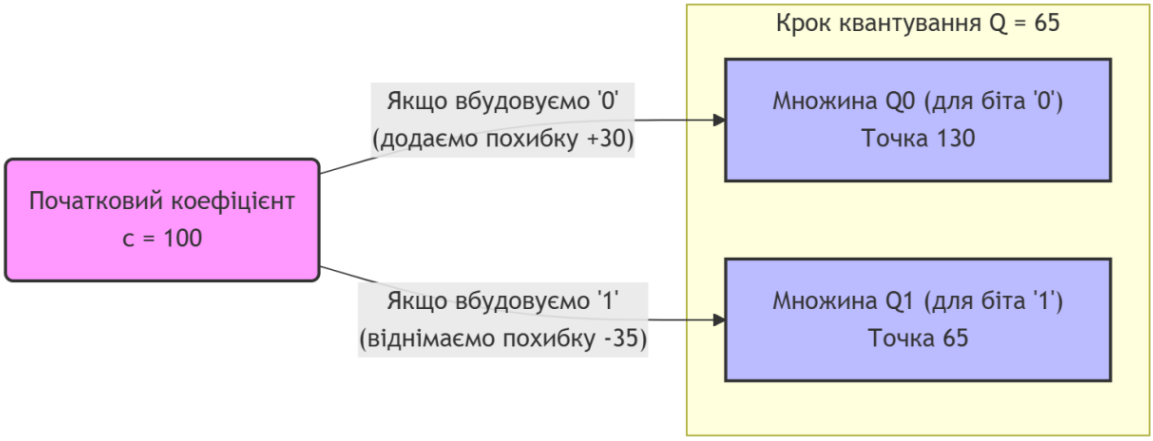


Рисунок 2.6 – Принцип роботи QIM для вбудовування водяного знака

Тобто ми подаємо цю операцію як деяке відображення Ф: $(B_{ij}, b_k) \to B'_{ij}$, яке вхідному блоку B_{ij} ставить у відповідність один біт b_k, що змінює лише один коефіцієнт DCT. Такий підхід "по одному блоку, по одному біту" навмисно обмежує щільність вбудовування в просторі, зменшуючи можливість появи спектральних артефактів. Зауважте, що велике значення Q = 65 обране не просто так: воно утворює досить широкий "захисний інтервал" (близько $\frac{Q}{2}$), котрий може витримувати шум обчислень зворотного квантування, втрати при черговому JPEG-квантуванні, а також агресивну обрізку в соцмережах, не створюючи помітного зниження якості PSNR.

Після зміни всіх необхідних коефіцієнтів, повністю заповнюємо послідовність B_R та виконуємо просторову реконструкцію зображення. Для кожного модифікованого блоку виконуємо обернене перетворення $B'_{ij} = IDCT(D'_{ij})$. Оскільки обернене перетворення (числа з плаваючою комою) не позбавлене дрібних помилок округлення просторової реконструкції (truncation error), отже, знову ж, для уникнення помилок, обробляємо результати "жорстким

відсіканням" (np.clip):

$$Y'(x,y) = clip\big(round\big(f'(x,y)\big), 0, 255\big), \quad (2.7)$$

де Y'(x,y) – фінальне скориговане цілочисельне значення пікселя каналу яскравості за просторовими координатами x та y, що повертається у просторовий домен зображення і є повністю готовим для збереження;

f'(x,y) – проміжне реконструйоване значення яскравості пікселя за координатами x та y після виконання зворотного дискретного косинусного перетворення (IDCT), яке до обробки може містити дробову частину або виходити за межі допустимого діапазону;

0 – задана нижня межа цільового діапазону (мінімально можливе значення інтенсивності для 8-бітного кодування пікселя, що відповідає відсутності світіння);

255 – задана верхня межа цільового діапазону (максимально можливе значення інтенсивності для 8-бітного кодування пікселя, що відповідає максимальному світінню);

x – просторова координата пікселя по горизонталі всередині оброблюваного матричного блоку;

y – просторова координата пікселя по вертикалі всередині оброблюваного матричного блоку.

Оновлений канал яскравості Y' конвертується у цілочисельний формат uint8, за допомогою функції cv2.merge об'єднується з незмінними кольоровими компонентами у структуру (Y', Cr, Cb), і перетворюється назад у стандартний колірний простір BGR.

На фінальному етапі відбувається експорт маркованого документа. Для збереження сумісності шляхів запису використовується кодування файлу у буфер пам'яті (cv2.imencode). Зображення кодується у формат JPEG з параметром якості $Q_{JPEG} = 100$. Цей крок встановлює режим, за якого мінімізується ризик

додаткового руйнування модифікованих DCT-коефіцієнтів матрицями стиснення безпосередньо під час генерації файлу. Такий підхід зберігає повний запас міцності алгоритму QIM для надійної протидії майбутнім атакам компресії при ймовірному аналоговому чи цифровому витоку.

Формалізуючи наведену логіку, інформаційну ємність (англ. payload capacity) розробленої системи можна описати як $C = zone_h \times zone_w$, рівень вбудовування (англ. embedding rate) як $ER = \frac{C}{n}$, а загальну обчислювальну складність алгоритму – приблизно як $O(hw) + O(N_{blocks})$. Узагальнено повний процес маркування подається як композиція послідовних операторів $I \to Y \to Z^* \to \{8 \times 8\} \to DCT \to QIM \to IDCT \to I'$. У функціональному вигляді це можна записати як:

$$I' = \Psi\big(I, M, Q, (3,2)\big) \tag{2.8}$$

де I' – результувальне марковане зображення (стеганоконтейнер), що містить приховану цифрову мітку та призначене для подальшого розповсюдження чи збереження;

Ψ – узагальнений математичний оператор функції вбудовування, який описує повний цикл стеганографічного перетворення (включаючи пряме дискретне косинусне перетворення, контентно-залежне маскування, модуляцію індексів квантування та зворотне перетворення);

I – оригінальне (вихідне) зображення, що виступає в ролі чистого контейнера до початку процесу маркування;

M – цільове службове повідомлення (сформований бітовий потік цифрового ідентифікатора співробітника), що підлягає приховуванню;

Q – глобальний крок квантування, який виступає ключем вбудовування та визначає баланс між стійкістю мітки і візуальною якістю (Q = 65);

(3,2) – фіксовані просторово-частотні координати цільового середньочастотного коефіцієнта всередині кожного матричного блоку 8×8 пікселів, обрані для мінімізації візуальних спотворень та стійкості до алгоритмів

компресії.

Таким чином, розроблений метод фактично пропонує дворівневий захист цифрової мітки, що базується на трьох надійних стовпах: Robustness = QIM + Content-Aware + Redundancy. Локальний захист спектра забезпечується стійким квантуванням індексів, а глобальний – вибором найскладнішої текстури і просторовим дублюванням. Такий підхід є унікальним, адже він дозволяє реалізувати сліпе вилучення (оригінал не потрібен), зберегти високу швидкість алгоритму, а також стійкість до JPEG-компресії.

Крім того, архітектура алгоритму повністю автономна і відповідає вимогам сліпих систем стеганографії, дозволяючи достовірно ідентифікувати джерело витоку конфіденційних даних без потреби звернення до оригінального файлу. При сліпому вилученні декодер паралельно сканує усі сегменти зображення: порожні ділянки генерують фільтрований білий шум, а обрана макро-зона видає чітку послідовність маркерів.

2.3 Розроблення алгоритму виявлення та зчитування мітки

Виявлення витоку інформації – це процес пошуку прихованого ідентифікатора в перехопленому зображенні. Одна з головних особливостей системи в цьому завданні – це сліпе вилучення (blind extraction), коли декодер може знайти та розшифрувати ЦВЗ без доступу до оригіналу або параметрів його створення.

Для зручності розгляду послідовність кроків вилучення і декодування мітки розділена на частини. Підготовка графічного контейнера і початок частотного сканування наведено у Додатку В, а завершення циклу вилучення і обробка результатів – у Додатку Г.

Робота модуля пошуку починається зі зчитування скомпрометованого файлу. Зображення завантажується як матриця пікселів за допомогою безпечного

побітового читання масиву байтів (numpy.fromfile) і подальшого декодування. Це дозволяє уникнути проблем із читанням файлів з кириличними назвами або розміщених у специфічних каталогах файлової системи Windows.

Після завантаження алгоритм аналізує структуру перехопленого документа. Якщо зображення кольорове, воно конвертується в яскравісно-колірний простір YCrCb, кольорові компоненти відкидаються, і аналіз продовжується по матриці яскравості Y. Якщо ж файл чорно-білий, алгоритм не буде конвертувати кольори, а працюватиме з оригінальною матрицею.

Щоб уникнути втрати точності при зворотних спектральних перетвореннях, робочий масив примусово конвертується у тип чисел з плаваючою комою (float32).

Наступним важливим кроком є просторове сканування графічного зображення. Під час вбудовування маркування система обчислювала дисперсію та обирала одну з найбільш текстурованих зон для приховування даних. Проте під час експертизи декодер працює в умовах повної невизначеності. Саме через те, що зловмисник може використовувати компресію JPEG, накладання фільтрів або кадрувати (кропувати) скриншот, параметри дисперсії перехопленого зображення можуть суттєво відрізнятися від оригінальних. Зважаючи на це, алгоритм уникає помилкової спроби вгадати початкову зону за допомогою повторного обчислення ознак текстури. Натомість матрицю яскравості віртуально розбивають на стандартну сітку 4×4, отримуючи 16 незалежних зон, і модуль починає глобальний цикл, проводячи "сліпе" детальне сканування кожної з них. Відмова від повторного вибору за дисперсією на користь глобального повнозонного сканування є неочевидною особливістю алгоритму та однією з причин його наукової новизни, оскільки система стає повністю незалежною від локальних геометричних чи текстурних спотворень контейнера.

Переходячи до будь-якої із зон, алгоритм виконує послідовний поблоковий аналіз. Подібно до етапу вбудовування, зону розглядають як набір незалежних блоків розміром 8×8 пікселів, для кожного із яких повторно обчислюють двовимірне дискретне косинусне перетворення. Отже, декодер переходить з

просторової області назад у частотну, де, власне, і маскується водяний знак. На відміну від традиційних схем цифрових водяних знаків, де для виявлення часто потрібне зіставлення з оригінальним спектром або використання збережених допоміжних ключів, декодер одразу аналізує заздалегідь зафіксований середньочастотний коефіцієнт DCT (3, 2), розглядаючи його як носій одного біта інформації. Оскільки вбудовування здійснювалося методом паритетного квантування, відновлювати початкове значення коефіцієнта не обов'язково. Алгоритм лише перевіряє, до якого квантувального інтервалу з кроком Q = 65 належить поточне значення коефіцієнта та яку парність воно має. Значення ділиться на величину кроку квантування та округлюється до цілого індексу, після чого обчислюється остача від ділення на 2. Парний індекс інтерпретується як біт 0, а непарний – як біт 1. Саме завдяки цій властивості метод є досить стійким: навіть якщо коефіцієнт суттєво зміститься після стиснення, він з великою часткою ймовірності все одно залишатиметься в межах свого широкого квантувального інтервалу та зберігатиме вбудований біт.

Послідовність бітів, прочитаних з усіх блоків зони, формує кандидатний бітовий потік можливої мітки, який відразу декодується у текстове подання шляхом групування в байти. Оскільки під час створення разом з інформацією у контейнер додатково вбудовували розділювач (###), декодер може автоматично визначати межі корисного навантаження, отже, довжину повідомлення зберігати не треба. Після перетворення потоку у текст виконується пошук цього маркера. Зони, що не містять маркування, генерують відфільтрований "білий шум", який автоматично відкидається. Для того, щоб виключити помилки при атрибуції через випадкову генерацію подібних бітових потоків, алгоритм звертається до додаткової лексичної перевірки. Кожний кандидат піддається тестуванню на відповідність очікуваній структурі: довжина більша за 3 символи та має включати розділювач | між ім'ям користувача та службовим кодом. Послідовності, що не відповідають цьому формату, відкидаються, що забезпечує додаткову, семантичну перевірку, окрім частотного виявлення.

Остаточне рішення приймається на основі статистичної обробки результатів

глобального сканування. Оскільки при вбудовуванні повідомлення було просторово продубльоване, часткові пошкодження зображення не призводять до знищення водяного знаку. Алгоритм використовує всі валідні фрагменти і з-поміж них проводить мажоритарне голосування (majority voting), за яким справжньою вважається така мітка, що зустрічається серед усіх кандидатів найбільшу кількість разів. Власне, це реалізує вбудований механізм корекції помилок без використання окремих, ресурсомістких кодів, що суттєво підвищує стійкість до атак. Принцип роботи механізму статистичної корекції помилок та відновлення цілісності ідентифікатора після деградації зображення подано на рисунку 2.7

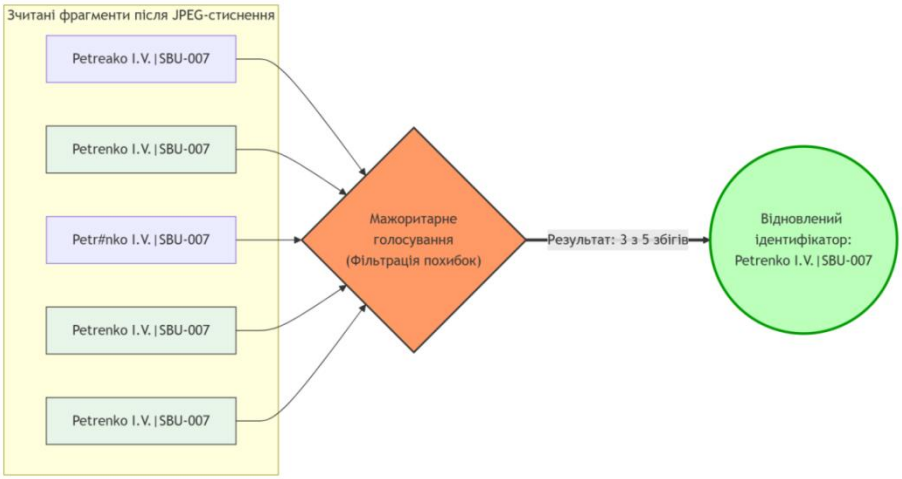


Рисунок 2.7 – Схема відновлення ідентифікатора методом мажоритарного голосування з пошкоджених фрагментів

Завдяки комбінації сліпого повнозонного сканування, застосування середньочастотних DCT-коефіцієнтів, просторового дублювання та мажоритарного голосування, описаний алгоритм функціонує після різних видів обробки, таких як компресія JPEG, послідовні повторні збереження, помірне зашумлення, згладжувальні фільтри, друк та подальше сканування, і навіть після часткового обрізання фрагментів документа.

Визначивши найімовірнішу мітку, система переходить до останнього етапу – атрибуції витоку. Декодований ідентифікатор виводиться оператору і

використовується для встановлення конкретного джерела компрометації.

У реальності повний конвеєр розслідування становить єдиний повний цикл: від завантаження контейнера і виділення яскравісної компоненти до повнозонного сканування, роботи з блоками, вилучення бітів, лексичної фільтрації і статистичного відновлення. Саме ця комбінація робить розглянутий алгоритм не тільки детектором водяних знаків, а повноцінним криміналістичним інструментом для надійного розслідування інцидентів внутрішньої безпеки.

2.4 Висновки до розділу 2

Розроблено та теоретично обґрунтовано алгоритми частотного цифрового маркування, які є математичним ядром прототипу системи ідентифікації джерела витоку конфіденційних графічних даних. На основі аналізу предметної області було розроблено алгоритм прихованого вбудовування та сліпого вилучення структурованих цифрових водяних знаків.

Обґрунтовано використання DCT та QIM. Показано, що перехід із просторової області у частотну відкриває можливість використання такої властивості дискретного косинусного перетворення, як ущільнення енергії (energy compaction), для відокремлення важливих деталей зображення від високочастотного шуму. Для вбудовування вибрано середньочастотний коефіцієнт DCT, що забезпечує оптимальний баланс між візуальною непомітністю мітки та стійкістю до деструктивних алгоритмів стиснення JPEG. Використання модуляції QIM, на відміну від адитивних схем, здійснює кодування шляхом визначення належності значення коефіцієнта до певного класу квантування, що робить маркування стійким до спотворень та шумів.

Визначено оптимальний колірний простір для вбудовування. Показано, що безпосереднє вбудовування даних у канали простору RGB є ненадійним і призводить до руйнування мітки. Натомість перетворення зображення у простір YCrCb та вбудовування мітки лише в канал яскравості (Y) виявилися найбільш

надійним підходом. Саме цей канал зберігає найбільшу енергетичну стабільність після JPEG-компресії, оскільки алгоритми стиснення з втратами цілеспрямовано відкидають інформацію із хроматичних компонентів (субдискретизація кольору).

Розроблено алгоритм адаптивного Content-Aware маскування. З метою мінімізації візуальних спотворень зображення віртуально розбивається на макро-зони, а вбудовування мітки проводиться лише в зоні із максимальною текстурною складністю, яка оцінюється за об'єктивним критерієм дисперсії пікселів.

Такий підхід реалізує концепцію перцептивного маскування (perceptual masking), коли складна текстура природного зображення статистично приховує внесені спектральні зміни від людського ока.

Інтегровано механізми просторової надмірності та корекції помилок. Згенерований бітовий потік із унікальним ідентифікатором користувача багаторазово дублюється в межах доступної площі обраної цільової зони зображення. Під час експертизи (декодування) застосовується метод мажоритарного голосування (majority voting), який відіграє роль механізму корекції помилок. Це потенційно гарантує виживання мітки та можливість її достовірного відновлення навіть після агресивного стиснення алгоритмами сучасних месенджерів чи соціальних мереж.

Запропоновано алгоритм повністю сліпого вилучення мітки (blind extraction). В основі розробленого алгоритму аналізу лежить відсутність необхідності наявності оригінального зображення чи використання специфічних ключів для його генерації. Декодер в свою чергу здійснює глобальне повнозонне сканування перехопленого файлу, лексичну фільтрацію кандидатних бітових потоків та їх статистичне відновлення. Даний конвеєр робить систему незалежною від локальних геометричних деформацій, часткового обрізання чи зашумлення документа зловмисником.

Отже, поєднання QIM, Content-Aware вибору зони та просторової надмірності формує надійний дворівневий захист цифрової мітки. Запропоновані алгоритми повною мірою відповідають сформованим вимогам щодо робастності та непомітності.

3 РЕАЛІЗАЦІЯ ТА ОЦІНКА ДОСТОВІРНОСТІ ПРОТОТИПУ СИСТЕМИ

3.1 Обґрунтування вибору засобів розробки

Для практичного підтвердження ефективності розроблених алгоритмів частотного цифрового маркування та сліпого вилучення мітки, які були теоретично обґрунтовані у другому розділі, необхідно здійснити їх програмну реалізацію. Вибір стека технологій обумовлюється специфікою поставлених завдань: необхідністю безперервної обробки багатовимірних матриць, наявністю оптимізованих функцій спектральних перетворень, підтримкою інтеграції з реляційними базами даних та можливістю побудови автономного модульного застосунку.

Як базова мова програмування була обрана мова Python (версії 3.14). Такий вибір обумовлений не лише розвиненою екосистемою наукових бібліотек, а й придатністю Python для швидкого прототипування складних алгоритмів цифрової обробки сигналів. Для задач стеганографічного маркування це є важливим, оскільки дослідницький процес потребує багаторазового тестування параметрів алгоритму (зокрема кроку квантування Q, координат цільового DCT-коефіцієнта та параметрів адаптивного вибору зон). Попри інтерпретовану природу Python, критичні до швидкодії математичні операції делегуються оптимізованим низькорівневим модулям, скомпільованим на мовах C та C++, що забезпечує високу обчислювальну ефективність при роботі з масивними графічними даними.

Фундаментальним ядром для виконання математичних обчислень у системі виступає бібліотека NumPy. Оскільки цифрові зображення представляють собою багатовимірні масиви пікселів, NumPy забезпечує векторизовані операції над такими структурами без використання ресурсомістких циклів рівня інтерпретатора. Зокрема, функціонал бібліотеки застосовується для віртуального розбиття матриці яскравості на макро-зони за допомогою механізму швидких зрізів (array slicing) та статистичного обчислення дисперсії (numpy.var) у кожному сегменті, що реалізує розроблений механізм контентно-залежного пошуку області вбудовування. Крім того, бібліотека забезпечує точну роботу з числами з

плаваючою комою (numpy.float32) та жорстке відсікання значень пікселів (numpy.clip) при просторовій реконструкції зображення, що дозволяє нівелювати обчислювальні похибки IDCT-перетворення і стабілізувати роботу модуля квантування QIM.

Основним інструментом комп'ютерного зору та спектральної обробки зображень обрано кросплатформну бібліотеку OpenCV (модуль cv2). Її роль у розробленому прототипі полягає у реалізації центрального спектрального ядра системи. Вибір OpenCV продиктований наявністю високою мірою оптимізованих вбудованих функцій прямого та зворотного дискретного косинусного перетворення (cv2.dct та cv2.idct), що дозволяє виконувати поблоковий спектральний аналіз практично в режимі реального часу. Це є надзвичайно важливим фактором на етапі криміналістичного сканування, де алгоритм має здійснити повнозонне (для 16 макро-областей) вилучення частотних коефіцієнтів перехопленого документа. Специфічною архітектурною особливістю застосування OpenCV у проєкті є відмова від стандартних функцій файлового вводу-виводу на користь методів кодування та декодування масивів байтів безпосередньо у пам'яті (cv2.imdecode, cv2.imencode). Такий підхід забезпечує коректну роботу з кириличними шляхами у файловій системі Windows та дозволяє жорстко встановлювати відсутність компресії на етапі генерації файлу ($Q_{JPEG} = 100$).

Для забезпечення вимог корпоративного сегмента та автоматизації процесу масової генерації маркованих копій у систему інтегровано модуль зв'язку з реляційною базою даних. Як СУБД обрано Microsoft SQL Server, оскільки вона є галузевим стандартом для зберігання облікових даних персоналу в державних та комерційних організаціях. Взаємодія з базою даних здійснюється за допомогою бібліотеки pyodbc, що реалізує надійне з'єднання через ODBC-драйвери. Це переводить прототип із рівня лабораторного алгоритму до концепції прикладної DLP-системи, дозволяючи динамічно підтягувати реальну інформацію про працівників (ПІБ, відділ, службовий код) і автоматизувати пакетну генерацію персоналізованих документів.

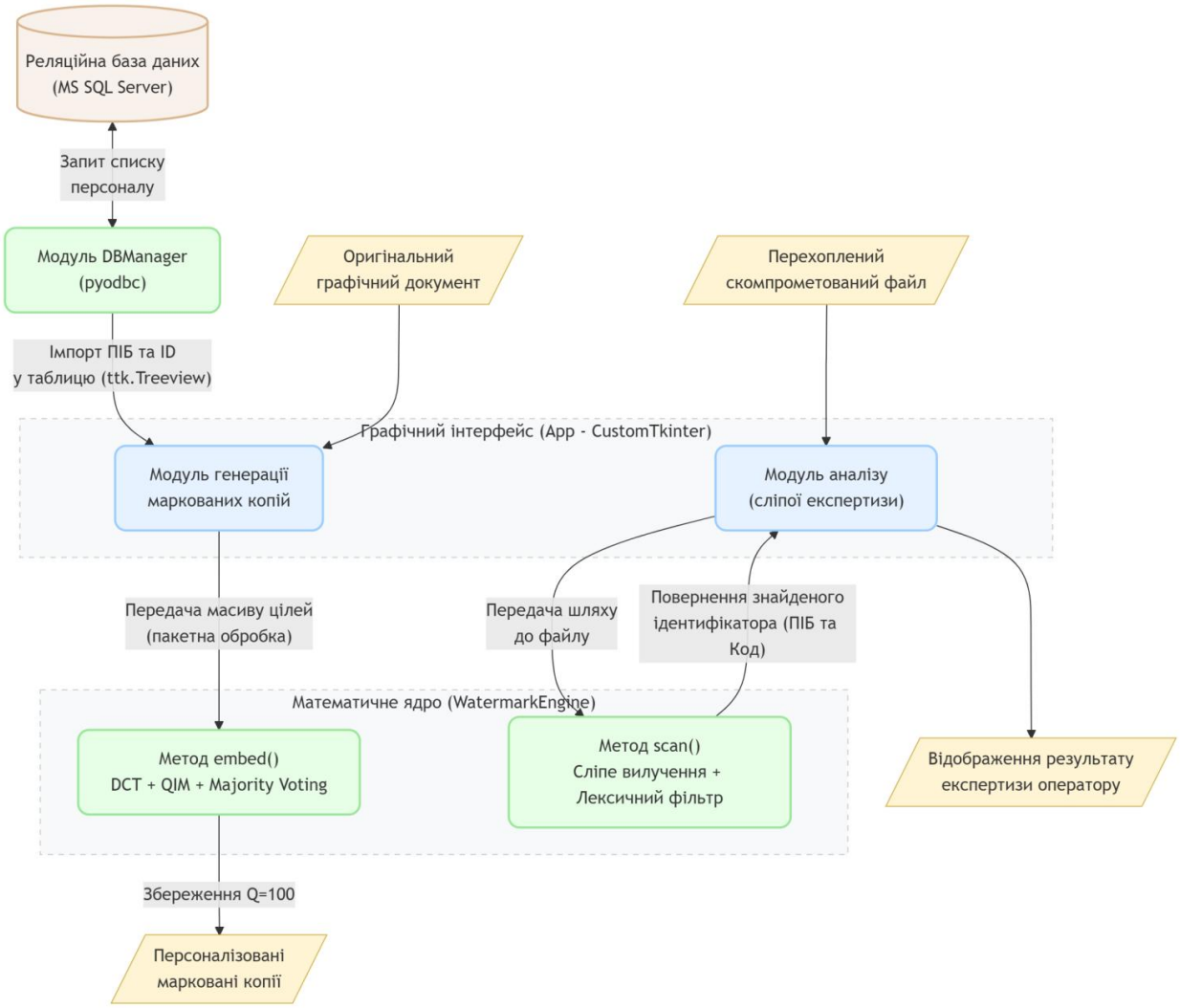
Щодо візуальної частини прототипу, для забезпечення зручності взаємодії оператора безпеки з математичним ядром системи розроблено графічний інтерфейс користувача (GUI) на базі бібліотеки customtkinter. Цей інструмент дозволив створити сучасний графічний інтерфейс, який логічно ізольований від математичної логіки, що дає змогу чітко розмежувати програмний комплекс на два функціональні контури: модуль пакетної генерації копій та модуль криміналістичного аналізу.

Отже, обраний стек технологій (Python, NumPy, OpenCV, pyodbc, CustomTkinter) повністю відповідає не лише базовим архітектурним вимогам, а й специфічним особливостям розробленого контентно-залежного DCT-QIM алгоритму. Він забезпечує ефективне виконання спектральних перетворень, пакетну генерацію персоналізованих копій та масштабованість прототипу до рівня повноцінної DLP-системи.

3.2 Реалізація алгоритмів у програмному середовищі

Процес створення прототипу системи ідентифікації джерела витоку графічних даних передбачав трансляцію розроблених математичних моделей контентно-залежного маркування (Content-Aware Watermarking), квантування індексів та DCT у структурований програмний код. Для забезпечення масштабованості та простоти подальшої підтримки, архітектуру програмного комплексу було побудовано за модульним принципом з використанням об'єктно-орієнтованого підходу. Таке проєктування дозволяє уникнути монолітної структури коду, де зміна одного компонента може призвести до критичних збоїв у роботі всієї системи.

Загальну структурну схему взаємодії основних компонентів системи (модуля генерації, бази даних та криміналістичного аналізу) наведено на рисунку 3.1.



3.1 – Схема взаємодії модулів програмного комплексу

Як видно зі схеми, логіка застосунку чітко розділена на три ізольовані компоненти: математичне ядро спектральних перетворень, модуль інтеграції з базою даних та графічний інтерфейс оператора. Такий підхід реалізує принцип слабкого зв’язування модулів (loose coupling) і дозволяє незалежно масштабувати або модифікувати окремі частини системи.

Усі криптографічні та спектральні обчислення інкапсульовані у статичному класі WatermarkEngine. Окремої уваги в його реалізації потребувало встановлення фіксованих параметрів алгоритму, оскільки вони безпосередньо впливають на баланс між непомітністю цифрової мітки та її стійкістю. Ці параметри винесені в глобальну конфігурацію ядра:

```
# ЛОГІКА (Core) - Content-Aware (Стабільна версія 4x4)
Q = 65
COEFF_X = 3
COEFF_Y = 2
DELIMITER = "###"
```

Глобальний крок квантування жорстко задається константою Q = 65. Вибір саме такого, порівняно великого значення, продиктований емпіричною необхідністю сформувати достатньо широкий квантувальний інтервал (decision boundaries). Цей інтервал слугує математичним буфером, який компенсує неминучі похибки округлення, що виникають під час зворотної просторової реконструкції (IDCT), а також поглинає шуми після агресивного JPEG-стиснення перехопленого документа в месенджерах. Аналогічно статично параметризовано координати цільового DCT-коефіцієнта – COEFF_X = 3 та COEFF_Y = 2. Константа DELIMITER відіграє критично важливу роль спеціального службового маркера, який автоматично конкатенується в кінець кожного повідомлення. Цей маркер забезпечує успішну лексичну фільтрацію на етапі сліпого вилучення, дозволяючи декодеру самостійно визначити межі корисного навантаження та відкинути випадковий спектральний шум.

Основний функціонал вбудовування цифрової мітки реалізовано у методі embed(). Розробка цього методу зіткнулася з проблемою системних кодувань ОС Windows, зокрема при читанні файлів із кириличними назвами або специфічними шляхами через стандартні функції (наприклад, cv2.imread). Для нівелювання цих проблем процес завантаження файлу було переписано через низькорівневе побітове читання у масив NumPy з подальшим його декодуванням в оперативній пам'яті. Після успішного завантаження зображення перетворюється у яскравісно-колірний простір YCrCb:

```
# Підтримка кирилиці
image_stream = np.fromfile(image_path, dtype=np.uint8)
image = cv2.imdecode(image_stream, cv2.IMREAD_COLOR)
if image is None: raise ValueError("Помилка читання файлу. Перевірте
формат або шлях.")
image_ycc = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)
Y, Cr, Cb = cv2.split(image_ycc)
```

```
Y = np.float32(Y)
h, w = Y.shape
```

Матриця каналу яскравості (Y) примусово приводиться до формату чисел із плаваючою комою (np.float32). Це є абсолютно необхідною умовою для підтримки високої точності обчислення спектральних частот, оскільки стандартний цілочисельний тип uint8 призвів би до катастрофічних втрат даних ще на етапі прямого DCT-перетворення.

Наступним кроком є реалізація механізму контентно-залежного маскування (Content-Aware). Замість глобального маркування всієї площі файлу, алгоритм віртуально розбиває контейнер на сітку макро-зон (у даній версії – 4x4) та за допомогою функції обчислення дисперсії визначає найскладнішу текстурну область. Цей підхід програмно імплементує біологічну концепцію адаптивного перцептивного маскування, приховуючи артефакти квантування у високочастотних змінах яскравості самого зображення:

```
h_blocks, w_blocks = h // 8, w // 8
grid_rows, grid_cols = 4, 4
zone_h = h_blocks // grid_rows
zone_w = w_blocks // grid_cols

best_zone = (0, 0)
max_variance = -1

for gr in range(grid_rows):
for gc in range(grid_cols):
y_start = gr * zone_h * 8
y_end = y_start + zone_h * 8
x_start = gc * zone_w * 8
x_end = x_start + zone_w * 8

zone_pixels = Y[y_start:y_end, x_start:x_end]
variance = np.var(zone_pixels)

if variance > max_variance:
max_variance = variance
best_zone = (gr, gc)
```

Важливим елементом архітектурної надійності розробленої системи стала програмна перевірка достатності місткості обраної цільової макро-зони. На

відміну від багатьох спрощених лабораторних реалізацій цифрової стеганографії, розроблений прототип превентивно блокує спробу вбудовування надмірно довгого тексту, що могло б викликати помилки типу «Index Out of Bounds» або призвести до часткового, неконсистентного запису повідомлення. Крім того, програмно реалізовано генерацію локальної просторової надмірності. Бітовий потік ідентифікатора багаторазово дублюється, помножуючись на обчислений коефіцієнт repeats. Цей підхід програмно імітує складні алгоритми помилкостійкого кодування (Forward Error Correction), значно підвищуючи загальну стійкість маркування:

```
zone_capacity = zone_h * zone_w
if len(base_bitstream) > zone_capacity:
raise ValueError(f"Текст занадто довгий для обраної зони (потрібно
{len(base_bitstream)} блоків, є {zone_capacity})!")

repeats = zone_capacity // len(base_bitstream)
if repeats == 0: repeats = 1
bitstream = base_bitstream * repeats
```

Безпосередня спектральна модифікація коефіцієнтів реалізується за допомогою контрольованого вкладеного циклу. Програма послідовно виокремлює блоки розміром 8x8 пікселів всередині знайденої цільової зони та застосовує до них дискретне косинусне перетворення. Механізм QIM (квантування індексів) імплементовано через логічну перевірку парності індексу щодо поточного цільового біта інформаційного потоку (target_bit). Якщо виявляється невідповідність, алгоритм коригує значення індексу на одиницю у бік найменшої математичної похибки. Після успішної модифікації блок піддається зворотному перетворенню (IDCT) і повертається у просторовий домен:

```
block = Y[i*8 : (i+1)*8, j*8 : (j+1)*8]
dct_block = cv2.dct(block)

coeff = dct_block[COEFF_X, COEFF_Y]
target_bit = int(bitstream[bit_index])

quantized = round(coeff / Q)
```

```
if quantized % 2 != target_bit:
if coeff >= quantized * Q: quantized += 1
else: quantized -= 1
dct_block[COEFF_X, COEFF_Y] = quantized * Q

Y[i*8 : (i+1)*8, j*8 : (j+1)*8] = cv2.idct(dct_block)
bit_index += 1
```

На фінальному етапі вбудовування модифікована матриця Y проходить процедуру жорсткого відсікання значень від 0 до 255 за допомогою функції np.clip. Це обов'язковий програмний крок, який усуває похибки просторової реконструкції, запобігаючи візуальним дефектам зображення (пересвітам чи інверсіям пікселів). Згодом оновлений канал яскравості зливається з хроматичними каналами, і готовий документ зберігається у пам'ять у форматі JPEG. Важливо зазначити, що для збереження запасу міцності алгоритму QIM, під час генерації встановлюється параметр повної відсутності втрат якості компресії (cv2.IMWRITE_JPEG_QUALITY, 100).

Процедура сліпого вилучення (blind extraction) реалізована у методі scan(). Вона не потребує оригінального файлу. Алгоритм виконує повнозонне сканування перехопленого зображення, вилучаючи біти за допомогою прямого DCT. Зібрані біти конвертуються у текст, після чого лексичний фільтр розділяє його за службовим маркером. Для відновлення мітки використовується мажоритарне голосування:

```
raw_text = WatermarkEngine.bits_to_text(zone_bits)
 parts = raw_text.split(DELIMITER)

# Фільтрація валідних ідентифікаторів
valid_parts = [p for p in parts if len(p) > 3 and p.count('|') >= 1]
all_extracted_texts.extend(valid_parts)

# Мажоритарне голосування
if all_extracted_texts:
counter = collections.Counter(all_extracted_texts)
best_match = counter.most_common(1)[0][0]
return True, best_match
```

Окремим архітектурним досягненням реалізованої системи є клас

DBManager, який перетворює алгоритм із суто теоретичного концепту на інструмент корпоративного рівня. Цей модуль відповідає за встановлення безпечного підключення до зовнішньої реляційної бази даних MS SQL Server за допомогою драйвера pyodbc. Програмна логіка передбачає динамічне підтягування реальної інформації про штат працівників (ПІБ, відділ та унікальний службовий код), що суттєво оптимізує та автоматизує подальший процес масового маркування. Для забезпечення високої експлуатаційної надійності (відмовостійкості), у методі роботи з базою даних реалізовано механізм перехоплення виключень try...except. Якщо підключення до серверної СКБД з технічних причин неможливе, модуль автоматично переходить у резервний режим і повертає масив безпечних тестових даних, не перериваючи роботу основного графічного інтерфейсу:

```
class DBManager:
@staticmethod
def get_personnel():
try:
conn = pyodbc.connect(CONNECTION_STRING)
cursor = conn.cursor()
cursor.execute("SELECT FullName, Department, ServiceCode FROM
Personnel")
rows = [list(row) for row in cursor.fetchall()]
conn.close()
return rows
except:
# Механізм відмовостійкості (fallback)
return [['Demo User', 'IT', '001'], ['Test User', 'HR', '002']]
```

Графічний інтерфейс (GUI) реалізовано через клас PolishedApp з використанням бібліотеки customtkinter. Інтерфейс підтримує пакетне маркування (run_batch) із залученням компонента ttk.Treeview, який відображає список співробітників. Метод циклічно обробляє обрані цілі та викликає ядро маркування:

```
def run_batch(self):
targets = [self.tree.item(i, "values") for i in
self.tree.get_children() if self.tree.item(i, "values")[0] == '[X]']
save_dir = filedialog.askdirectory()
```

```
cnt = 0
for t in targets:
name = str(t[1]).replace("'", "").replace("(", "")
code = str(t[3]).replace("'", "").replace(")", "").replace(",", "")
text = f"{name}|{code}"
fname = f"document_{code}.jpg"

WatermarkEngine.embed(self.encode_path, text, os.path.join(save_dir,
fname))
cnt += 1
```

Таким чином, розроблений програмний код повністю відтворює архітектуру теоретичних моделей і формує надійний, об'єктно-орієнтований прототип системи захисту інформації.

3.3 Експериментальна перевірка достовірності системи

Завершальним етапом розробки програмного прототипу системи ідентифікації джерела витоку є проведення комплексної експериментальної перевірки. Метою тестування є об'єктивне підтвердження того, що реалізований контентно-залежний алгоритм на базі DCT та QIM повною мірою задовольняє ключовим критеріям систем класу Data Loss Prevention (DLP): абсолютній візуальній непомітності вбудованої мітки та її високій робастності (стійкості) до деструктивних впливів під час пересилання через популярні цифрові канали зв'язку.

Експериментальне дослідження проводилося у два послідовні етапи:

– оцінка візуальної якості маркованого зображення-контейнера за допомогою обчислення об'єктивних математичних метрик;

– стрес-тестування алгоритму сліпого вилучення мітки в умовах агресивної компресії JPEG та моделювання реального витоку через месенджери.

Відповідно до вимог, сформованих на початку дослідження, вбудовування ідентифікатора співробітника не повинно викликати візуальних артефактів, розмиття або локальних змін контрастності, які могли б бути помічені

неозброєним оком або спонукати інсайдера до застосування додаткових засобів графічної фільтрації.

Для об'єктивної оцінки рівня спотворень після вбудовування водяного знака (із застосуванням встановленого кроку квантування Q = 65) було розроблено допоміжний програмний інструмент – "Diploma Tool: Image Quality Assessment". Даний модуль автоматизує обчислення метрики пікового відношення сигналу до шуму (Peak Signal-to-Noise Ratio, PSNR) та середньоквадратичної похибки (Mean Squared Error, MSE).

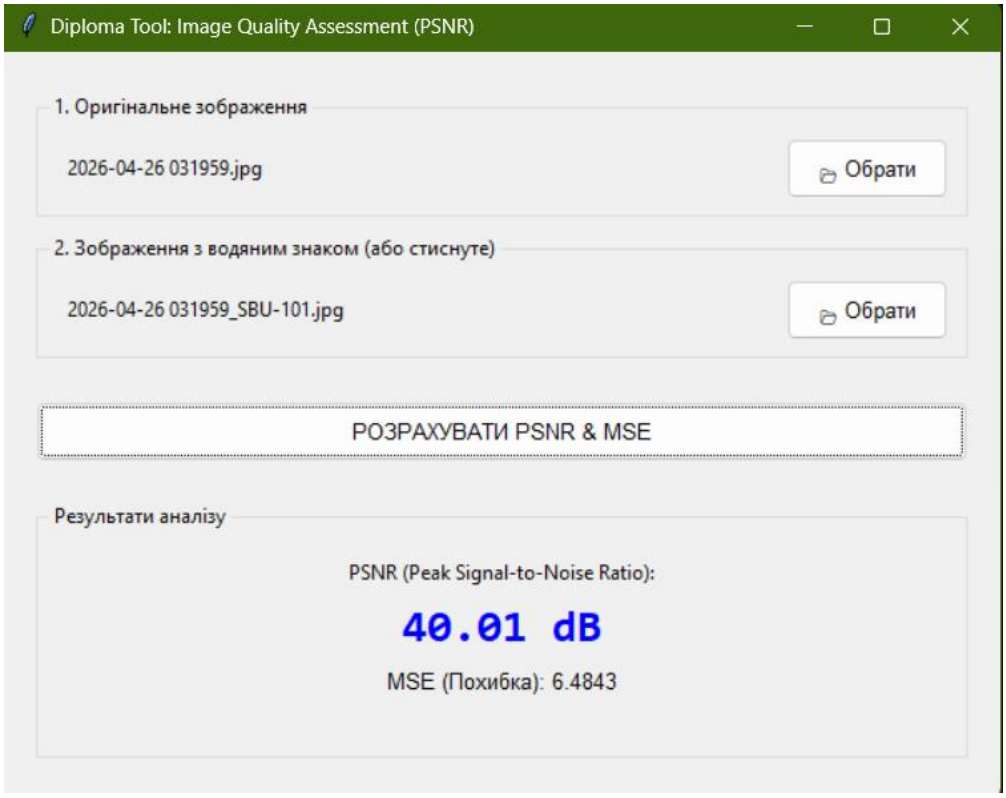


Рисунок 3.2 – Результат обчислення метрик візуальної якості маркованого документа

Як видно з результатів аналізу, наведених на рисунку 3.2, для тестового скриншота документа показник PSNR склав 40.01 dB, при цьому середньоквадратична похибка (MSE) дорівнює 6.4843. У теорії цифрової стеганографії значення PSNR, що перевищують поріг у 35 дБ, вважаються такими, що гарантують візуальну ідентичність зображень. Відповідно, отриманий

результат (понад 40 дБ) беззаперечно свідчить про те, що реалізований механізм адаптивного перцептивного маскування (Content-Aware) повністю виконав своє завдання. Зміна середньочастотного коефіцієнта F(3,2) у високотекстурних областях успішно приховує спектральні артефакти від людської візуальної системи.

Найбільш критичною загрозою для прихованих даних є алгоритми стиснення з втратами (Lossy Compression), які застосовуються за замовчуванням у більшості сучасних соціальних мереж та месенджерів. Для перевірки надійності розробленого прототипу було змодельовано ситуацію реального "аналогового витоку": маркований документ був завантажений та пересланий через десктопну версію месенджера Telegram. Відомо, що даний застосунок автоматично застосовує власні агресивні алгоритми оптимізації та ресайзингу фотографій для економії мережевого трафіку. Після пересилання скомпрометований файл був завантажений і переданий до модуля аналізу розробленої системи.

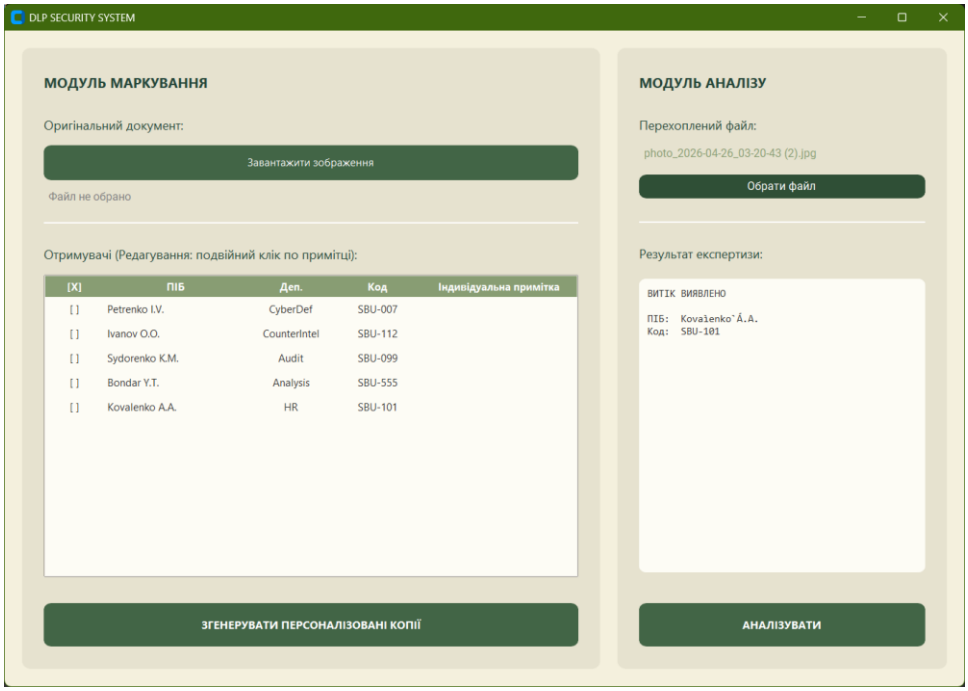


Рисунок 3.3 – Результат експертизи документа після пересилання через месенджер Telegram

Результати експертизи (рисунок 3.3) яскраво демонструють високу

практичну стійкість обраного математичного апарату. Модуль сліпого вилучення успішно просканував файл, локалізував мітку та декодував службовий код (SBU-101) і ПІБ порушника. Спостерігається поява незначного текстового артефакту (зміна літери на апостроф у рядку Kovaìenko'Á.A.), що є цілком очікуваним наслідком часткової втрати спектральних даних після стиснення. Проте ці артефакти не руйнують загальної семантики повідомлення, і прізвище залишається абсолютно читабельним для оператора безпеки, що дозволяє швидко та безпомилково ідентифікувати інсайдера.

Для визначення граничних меж міцності алгоритму було проведено додаткове стрес-тестування. Зображення, що вже зазнало деградації у Telegram, було піддано додатковій примусовій компресії у форматі JPEG зі зниженням якості до 70%.

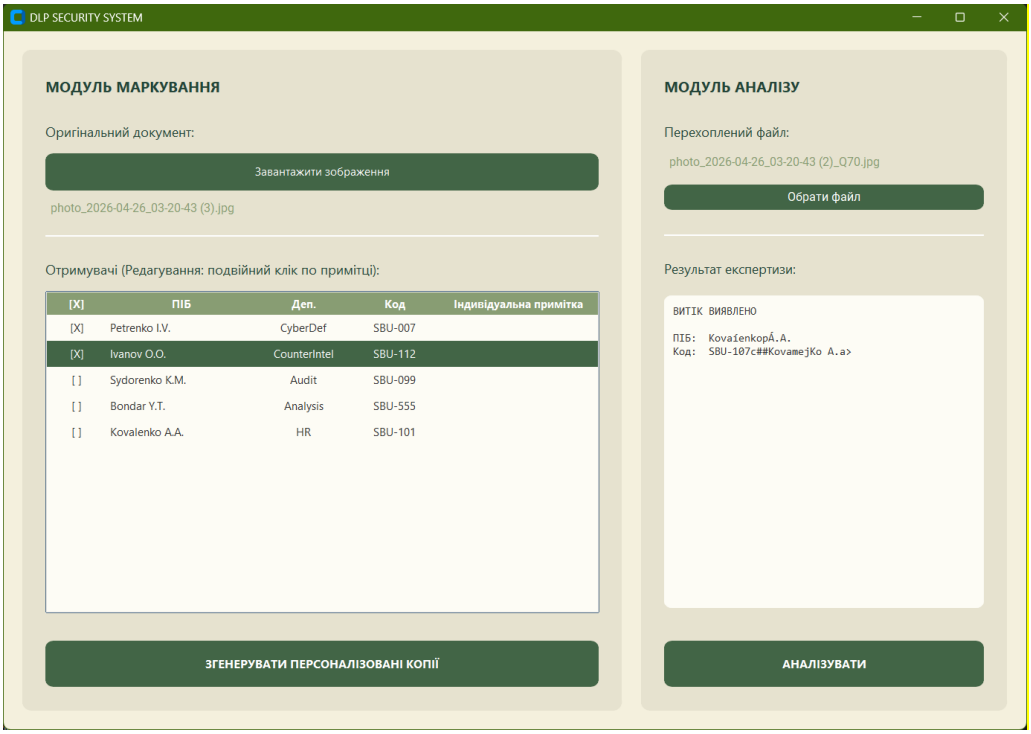


Рисунок 3.4 – Результат експертизи після додаткового екстремального стиснення (JPEG 70%)

Як видно з рисунка 3.4, екстремальне подвійне стиснення призводить до суттєвої втрати високо- та середньочастотних коефіцієнтів матриці DCT.

Кількість артефактів та нерозпізнаних символів у декодованому рядку значно збільшується (KovaíenkopÁ.A., SBU-107c##KovamejKo A.a>). Проте ключовою перевагою алгоритму є використання методів просторової надмірності (багаторазове дублювання потоку) та мажоритарного голосування під час сканування. Завдяки цим механізмам ключовий ідентифікатор (прізвище) виживає навіть у настільки екстремальних умовах, що робить систему придатною для розслідування реальних інцидентів.

Для комплексної та глибокої оцінки процесу деградації цифрової мітки було побудовано графік залежності коефіцієнта бітових помилок (Bit/Character Error Rate, BER) від рівня якості збереження JPEG. Цей показник обчислювався за допомогою бібліотеки difflib як відношення кількості спотворених символів до загальної довжини оригінального повідомлення на кожному кроці компресії (від 100% до 50% з кроком у 5%).

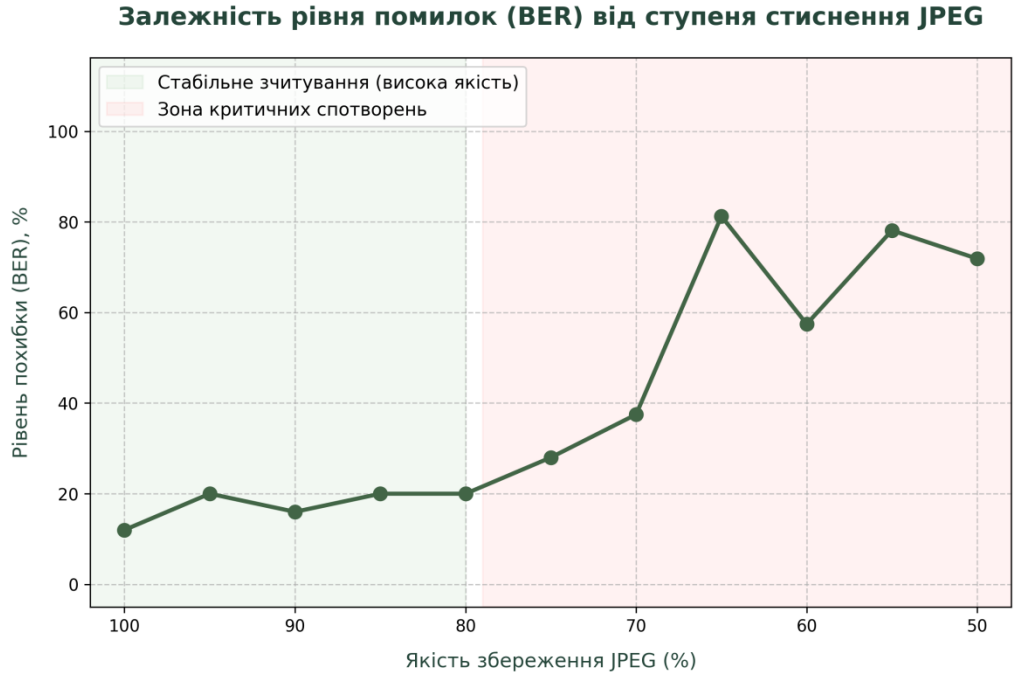


Рисунок 3.5 – Графік залежності рівня помилок (BER) від ступеня стиснення зображення

Аналіз отриманого графіка (рисунок 3.5) дозволяє зробити такі обґрунтовані

висновки щодо достовірності та надійності роботи системи:

– зона стабільного зчитування (JPEG 100% – 80%): При збереженні зображення з високою якістю алгоритми компресії практично не зачіпають цільовий коефіцієнт F(3,2). Показник похибки у цій зоні мінімальний (від 10% до 20%, що відповідає появі 2-4 артефактних символів на весь рядок). Це гарантує безпомилкове сліпе вилучення загальної семантики мітки.

– зона появи критичних спотворень (JPEG 79% – 50%): Зниження якості нижче 80% призводить до обнулення середніх частот. Рівень помилок починає стрімко зростати (від 30% до 80%), досягаючи максимуму при якості нижче 65%. У таких умовах мітка зазнає сильної руйнації. Однак варто зауважити, що візуальна якість самого документа при стисненні нижче 60% деградує настільки сильно (текст розмивається і стає нечитабельним), що контейнер втрачає свою інформаційну цінність для потенційного зловмисника.

Підсумовуючи результати проведених тестів, можна стверджувати, що розроблена система цифрового маркування демонструє високий рівень достовірності. Прототип успішно довів свою ефективність, задовольнивши вимоги як до візуальної непомітності (PSNR > 40 дБ), так і до практичної робастності при пересиланні конфіденційних даних через неконтрольовані канали зв'язку.

Окремим, але вкрай важливим вектором експериментального дослідження стала перевірка роботи алгоритму на фотографіях документів, зроблених на камеру смартфона. Незважаючи на те, що розроблена система першочергово проєктувалася для захисту цифрових скриншотів та плоских скан-копій (де матриця пікселів не має геометричних спотворень), метод контентно-залежного вбудовування продемонстрував високу ефективність і на зашумлених фотографіях фізичних носіїв.

Суб'єктивний візуальний аналіз показав, що на фотографіях документа цифрова мітка стає навіть менш помітною, ніж на цифрових оригіналах. Це явище має чітке наукове обґрунтування: природний оптичний шум матриці камери смартфона та мікротіні від нерівномірного освітлення виступають ідеальною

додатковою перцептивною маскою, у якій повністю розчиняються спектральні артефакти QIM. Об'єктивні метрики підтверджують цю тезу: під час тестування фотографії документа показник PSNR склав 45.92 дБ (рисунок 3.6).

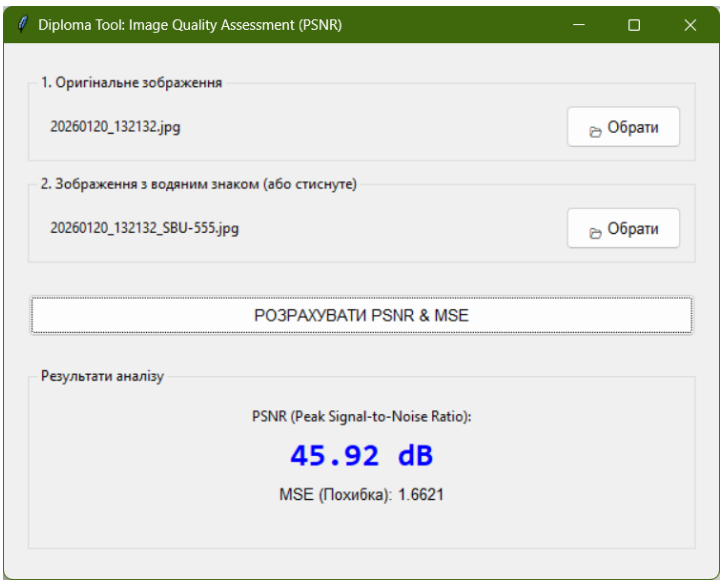


Рисунок 3.6 – Оцінка візуальної якості фотографії документа після вбудовування мітки

Для наочної демонстрації абсолютної візуальної непомітності маркування наведено порівняння фрагментів фотографії до та після застосування розробленого алгоритму:

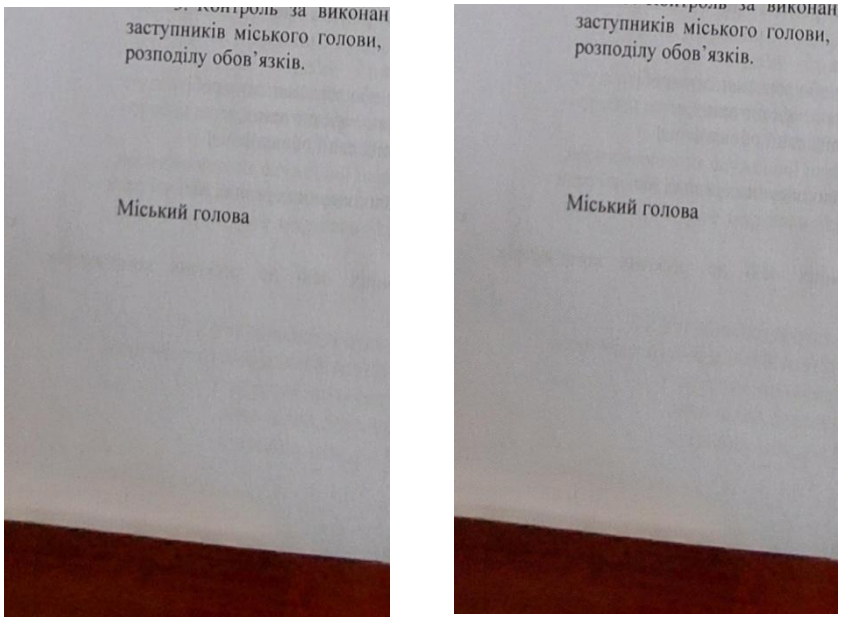


Рисунок 3.7 – Порівняння фрагментів фотографії документа: зліва – оригінал, справа – маркована копія

Для більш детального аналізу характеру спектральних змін, внесених алгоритмом, доцільно розглянути багаторазово збільшені (масштабовані) фрагменти цільових текстурних зон.

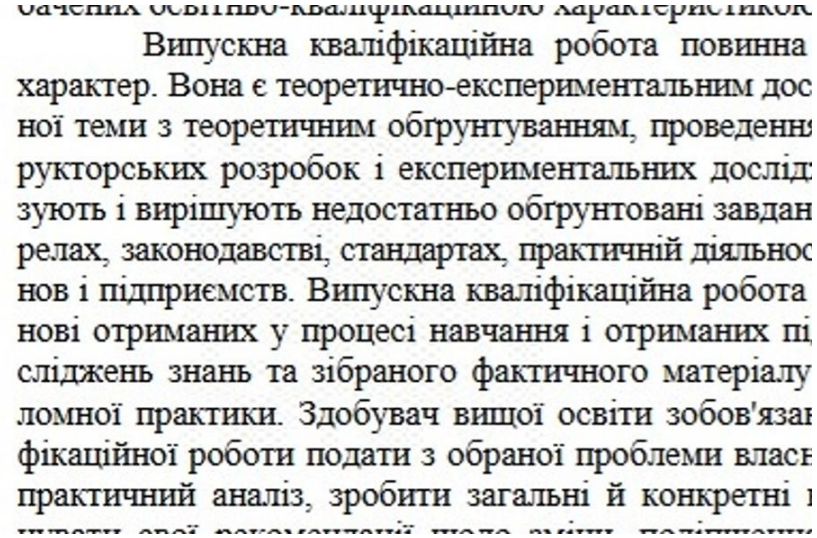


Рисунок 3.8 – Багаторазово збільшений фрагмент маркованого цифрового скриншота

Як видно з рисунка 3.8, на ідеально чистому фоні цифрового скриншота при екстремальному наближенні можна помітити незначний високочастотний шум у вигляді характерної "блокової" текстури. Це є прямим візуальним наслідком квантування середніх частот алгоритмом QIM.

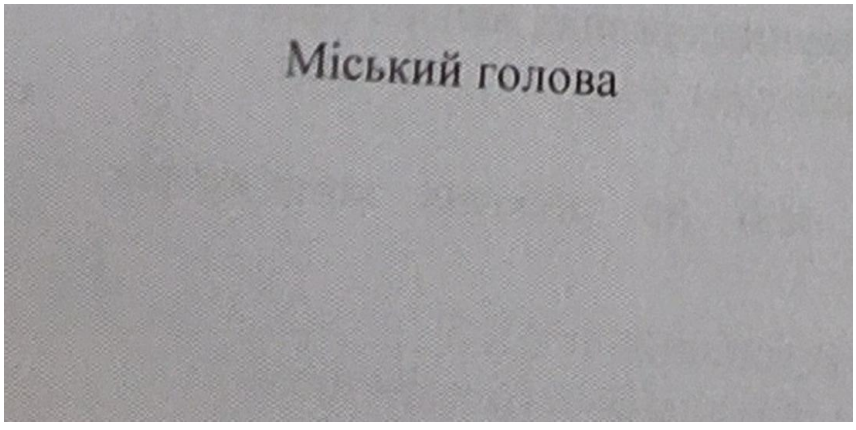


Рисунок 3.9 – Багаторазово збільшений фрагмент маркованої фотографії документа

Натомість на збільшеному фрагменті фотографії (рисунок 3.9) цей самий штучний спектральний шум повністю та органічно зливається з природним зерном (шумом ISO) матриці смартфона та оптичними нерівностями освітлення паперу. Це наочно пояснює, чому вбудована мітка в умовах фотографування стає абсолютно невідрізненною від звичайних артефактів камери, що й підтверджується вищим показником PSNR.

Проте варто зазначити, що якщо етап вбудовування мітки у фотографії працює бездоганно, то процес її сліпого вилучення (детекції) стикається з новими математичними викликами. При фотографуванні екрана або аркуша паперу під кутом неминуче виникають геометричні спотворення (перспективні викривлення, обертання, масштабування). Оскільки розроблений криміналістичний сканер оперує жорсткою сіткою DCT-блоків 8x8 пікселів, такі афінні перетворення можуть змістити частотні коефіцієнти, ускладнюючи пряме зчитування. З огляду на це, розробка додаткового модуля попередньої геометричної нормалізації зображення (наприклад, на основі комп'ютерного зору для пошуку кутів документа та їх вирівнювання) є надзвичайно перспективним напрямом для подальших досліджень, який дозволить зчитувати мітки з фотографій так само ефективно, як і зі скриншотів.

Підсумовуючи результати всіх етапів тестування, можна констатувати, що розроблений програмний прототип повністю виконав поставлені завдання. Система продемонструвала здатність генерувати візуально непомітні мітки (PSNR > 40 дБ), гарантовано ідентифікувати порушника після проходження документа через агресивні фільтри месенджерів (зокрема Telegram) та витримувати додаткове JPEG-стиснення до рівня 70% із подальшим успішним мажоритарним відновленням семантики повідомлення.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано науково-прикладне завдання з розроблення прототипу системи ідентифікації джерела витоку графічних даних на основі частотного цифрового маркування. Впровадження результатів роботи дозволяє суттєво підвищити рівень інформаційної безпеки та забезпечити невідворотність відповідальності за компрометацію даних через «аналогові» канали витоку.

В результаті проведеної роботи сформульовано наступні загальні висновки:

– обґрунтовано вибір методів. Аналіз довів вразливість класичних DLP-систем до фізичного захоплення даних (фотографування, скриншоти). На відміну від нестійких просторових методів, обрана комбінація DCT та QIM забезпечила оптимальний баланс між візуальною непомітністю та робастністю до стиснення;

– розроблено адаптивні алгоритми. Впроваджено контентно-залежний підхід, що автоматично обирає для маркування зони з максимальною текстурною складністю, реалізуючи ефект перцептивного маскування. Використання каналу яскравості Y, модифікація коефіцієнта F(3,2) з кроком Q=65 та просторове дублювання бітового потоку сформували надійний механізм захисту;

– створено модуль сліпого вилучення. Розроблений алгоритм детекції не потребує наявності оригінального зображення. Завдяки повнозонному скануванню, лексичній фільтрації та механізму мажоритарного голосування (majority voting), система здатна достовірно відновлювати ідентифікатор порушника навіть за наявності локальних бітових помилок;

– здійснено програмну реалізацію. Створено повнофункціональний модульний прототип DLP-системи на мові Python (OpenCV, NumPy) з інтеграцією до БД MS SQL Server. Це забезпечило автоматизацію пакетної генерації персоналізованих копій документів для корпоративного використання;

– стрес-тестування робастності алгоритму підтвердило його надзвичайну стійкість до агресивного алгоритмічного стиснення. Система безпомилково ідентифікувала джерело витоку після пересилання маркованого документа через

месенджер Telegram. Аналіз залежності коефіцієнта бітових помилок від якості JPEG-компресії показав, що алгоритм зберігає здатність до мажоритарного відновлення семантики повідомлення навіть при екстремальному зниженні якості збереження до 70%;

– встановлено високу перспективність застосування алгоритму для захисту фізичних носіїв інформації. Експериментально доведено, що під час створення фотокопій документа на камеру смартфона, природне зерно матриці (ISO) виступає додатковою перцептивною маскою, підвищуючи показник візуальної якості до 45,92 дБ.

Підсумовуючи, можна стверджувати, що розроблена система цифрового маркування є дієвим інструментом розслідування інцидентів кібербезпеки. Практичне застосування результатів кваліфікаційної роботи в існуючих системах електронного документообігу дозволить значно знизити ризики інсайдерських витоків інформації. Подальший розвиток розробленого прототипу доцільно спрямувати на інтеграцію модулів комп’ютерного зору для попередньої геометричної нормалізації фотографій (усунення афінних викривлень), що забезпечить ще вищу точність детекції цифрових водяних знаків із фізичних копій документів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Le D. C., Zincir-Heywood N. Exploring anomalous behaviour detection and classification for insider threat identification. International Journal of Network Management. 2021. Vol. 31, no. 4. P. 2109.

2. Cost of a Data Breach Report 2025 : Technical Report [Електронний ресурс] / IBM. Armonk, NY, USA, 2025. URL: https://bakerdonelson.com/webfiles/Publications/20250822_Cost-of-a-Data-Breach-Report-2025.pdf (дата звернення: 05.04.2026).

3. Cost of a Data Breach Report 2024 : Technical Report [Електронний ресурс] / IBM. Armonk, NY, USA, 2024. (дата звернення: 11.04.2026).

4. A comprehensive systematic literature review on intrusion detection systems / M. Ozkan-Okay, R. Samet, Ö. Aslan, D. Gupta. IEEE Access. 2021. Vol. 9. P. 157727–157760.

5. Advancements and Best Practices in Data Loss Prevention: A Comprehensive Review / N. Abid et al. Journal of Information Security and Applications. 2024.

6. Bypassing DLP's – The Analog Gap [Електронний ресурс]. Levent Durdalı's Blog. URL: https://omerwwazap.github.io/blog/posts/BypassingDLP/ (дата звернення: 15.04.2026).

7. Khan S., Saleem M. A., Akbar M. Anomaly Detection and Enterprise Security using User and Entity Behavior Analytics (UEBA). 2022 3rd International Conference on Innovations in Computer Science & Software Engineering (ICONICS). IEEE, 2022.

8. TabSec: A Collaborative Framework for Novel Insider Threat Detection / Y. Wang et al. IEEE Access. 2024. Vol. 12.

9. Digital Watermarking: Types and Importance of Digital Watermarking [Електронний ресурс]. Zero Trust Blog. URL: https://instasafe.com/blog/digital-watermarking-and-its-types/ (дата звернення: 22.04.2026).

10. Digital Watermarking and its Types [Електронний ресурс]. GeeksforGeeks. URL: https://www.geeksforgeeks.org/computer-networks/digital-watermarking-and-its-types/ (дата звернення: 10.04.2026).

11. Gao G., Xu T., Hua F. Robust Image Watermarking Based on Generative Adversarial Networks for Copyright Protection. 2024.

12. San C. T. Spatial Vs Frequency Domain: A Guide to Image Interpretation [Електронний ресурс]. Medium. URL: https://medium.com/@chawthirisan/spatial-vs-frequency-domain-a-guide-to-image-interpretation-d9c16b129b3f (дата звернення: 04.04.2026).

13. Spatial Domain Method – an overview [Електронний ресурс]. ScienceDirect Topics. URL: https://www.sciencedirect.com/topics/engineering/spatial-domain-method (дата звернення: 16.04.2026).

14. Analogue to Digital Converter (ADC) Basics [Електронний ресурс]. Electronics Tutorials. URL: https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html (дата звернення: 09.04.2026).

15. Understanding LSB (Least Significant Bit) in Analog-to-Digital Conversion [Електронний ресурс]. DigiKey TechForum. URL: https://forum.digikey.com/t/understanding-lsb-least-significant-bit-in-analog-to-digital-conversion/59573 (дата звернення: 25.04.2026).

16. Spatial domain processing [Електронний ресурс]. Fiveable. URL: https://fiveable.me/images-as-data/unit-3/spatial-domain-processing/study-guide/cKAa7Z4sA6Fq3Kja (дата звернення: 12.04.2026).

17. Cedillo-Hernandez M., Cedillo-Hernandez A., Garcia-Ugalde F. J. Improving DFT-Based Image Watermarking Using Particle Swarm Optimization Algorithm. Mathematics. 2021. Vol. 9. P. 1795. URL: https://doi.org/10.3390/math9151795.

18. Dubey N., Modi H. A Robust Discrete Wavelet Transform Based Adaptive Watermarking Scheme in YCbCr Color Space against Camcorder Recording in Cinema/Movie Theatres [Електронний ресурс]. URL: https://espublisher.com/uploads/article_pdf/es8d491.pdf (дата звернення: 26.04.2026).

19. Discrete Cosine Transform [Електронний ресурс]. MATLAB & Simulink. URL: https://www.mathworks.com/help/images/discrete-cosine-transform.html (дата звернення: 09.04.2026).

20. Wu W., Dong Y., Wang G. Image Robust Watermarking Method Based on

DWT-SVD Transform and Chaotic Map. Complexity. 2024. P. 6618382. URL: https://doi.org/10.1155/2024/6618382.

21. Stathaki T. Digital Image Processing Image Transforms: The 2D Discrete Cosine Transform [Електронний ресурс]. URL: http://commsp.ee.ic.ac.uk/~tania/teaching/DIP%202014/DIP%20DCT%202019.pdf (дата звернення: 21.04.2026).

22. Two-Dimensional Discrete Cosine Transform – an overview [Електронний ресурс]. ScienceDirect Topics. URL: https://www.sciencedirect.com/topics/computer-science/two-dimensional-discrete-cosine-transform (дата звернення: 07.04.2026).

23. Product Documentation – NI. NI-Test- und Messlösungen von Emerson [Електронний ресурс]. NI. URL: https://www.ni.com/docs/en-US/bundle/labview-api-ref/page/vi-lib/analysis/2dsp-llb/inverse-dct-vi.html#d759525e120 (дата звернення: 26.04.2026).

24. Two-dimensional inverse discrete cosine transform using SIMD instructions : US6907438B1 [Електронний ресурс]. Google Patents. URL: https://patents.google.com/patent/US6907438B1/en (дата звернення: 14.04.2026).

25. Content-Aware Quantization Index Modulation: Leveraging Data Statistics for Enhanced Image Watermarking / J. Mao et al. IEEE Transactions on Information Forensics and Security. 2023. P. 1–1. URL: https://doi.org/10.1109/TIFS.2023.3342612.

26. Alobaidi T. Rate-Distortion Optimized Quantization Index Modulation in Robust Image Watermarking. Scientific Research Journal of Engineering and Computer Sciences. 2026. Vol. 6. P. 1–9. URL: https://doi.org/10.47310/srjecs.2026.v06i01.001.

27. Zhang J. Understanding DCT and Quantization in JPEG compression [Електронний ресурс]. DEV Community. URL: https://dev.to/marycheung021213/understanding-dct-and-quantization-in-jpeg-compression-1col (дата звернення: 07.04.2026).

28. Wang Y., Yang C., Ding K. Multiple Watermarking Algorithms for Vector Geographic Data Based on Multiple Quantization Index Modulation. Applied Sciences. 2023. Vol. 13. P. 12390. URL: https://doi.org/10.3390/app132212390.

29. Gaur S., Barthwal V. An Extensive Analysis of Digital Image Watermarking

Techniques.

30. A Comparative Study of Various Digital Image Watermarking Techniques: Specific to Hybrid Watermarking.

31. Gao G., Xu T., Hua F. Robust image watermarking based on generative adversarial networks for copyright protection. 2024.

32. Resampling-Detection-Network-Based Robust Image Watermarking against Scaling and Cutting / H.-L. Li, X.-Q. Zhang, Z.-H. Wang, Z.-M. Lu, J.-L. Cui. Sensors. 2023. Vol. 23. P. 8195. URL: https://doi.org/10.3390/s23198195.

33. Edge-Aware Dual-Task Image Watermarking Against Social Network Noise / H. Jiang et al. Applied Sciences. 2025. Vol. 15. P. 57. URL: https://doi.org/10.3390/app15010057.

34. Tcheslavski G. V., Vasefi M. An “Instantaneous” Response of a Human Visual System to Hue: An EEG-Based Study. Sensors. 2022. Vol. 22. P. 8484. URL: https://doi.org/10.3390/s22218484.

35. Optimal Implementation of Dynamical Visual Cryptography Scheme for Imaging-Based Testing of Human Visual System / L. Saunoriene, P. Palevicius, A. Gelzinis, M. Ragulskis. Mathematics. 2026. Vol. 14. P. 1020. URL: https://doi.org/10.3390/math14061020.

36. Integral Imaging Display System Based on Human Visual Distance Perception Model / L. Deng, Z. Li, Y. Gu, Q. Wang. Sensors. 2023. Vol. 23. P. 9011. URL: https://doi.org/10.3390/s23219011.

37. A Robust and Secure Watermarking Approach Based on Hermite Transform and SVD-DCT / S. L. Gomez-Coronel et al. Applied Sciences. 2023. Vol. 13. P. 8430. URL: https://doi.org/10.3390/app13148430.

38. Deep Reinforcement Learning-Based DCT Image Steganography / R. Yang, L. Liu, B. Han, F. Hu. Mathematics. 2025. Vol. 13. P. 3150. URL: https://doi.org/10.3390/math13193150.

39. Li F., Wang Z. A Zero-Watermarking Algorithm Based on Vortex-like Texture Feature Descriptors. Electronics. 2024. Vol. 13. P. 3906. URL: https://doi.org/10.3390/electronics13193906.

40. Covert Communication through Robust Fragment Hiding in a Large Number of Images / P. Wang et al. Sensors. 2024. Vol. 24. P. 627. URL: https://doi.org/10.3390/s24020627.

41. A JPEG Reversible Data Hiding Algorithm Based on Block Smoothness Estimation and Optimal Zero Coefficient Selection / Y. Yue, M. Zhang, P. Lai, F. Di. Applied Sciences. 2025. Vol. 15. P. 10282. URL: https://doi.org/10.3390/app151810282.

42. Adaptive Code-Controlled Steganography with Enhanced Robustness to JPEG Compression / N. Kazakova et al. Symmetry. 2026. Vol. 18. P. 632. URL: https://doi.org/10.3390/sym18040632.

43. Ouyang J., Wang R., Shi T. Robust Watermarking Algorithm Based on QGT and Neighborhood Coefficient Statistical Features. Electronics. 2025. Vol. 14. P. 4494. URL: https://doi.org/10.3390/electronics14224494.

44. Salane N. A DCT-Based Image Watermarking Scheme with Geometric Invariance and Redundant Embedding. 2026. URL: https://doi.org/10.13140/RG.2.2.26477.55523.

45. Bao B., Wang Y. A robust blind color watermarking algorithm based on the Radon-DCT transform. Multimedia Tools and Applications. 2024. Vol. 83. P. 1–20. URL: https://doi.org/10.1007/s11042-023-17875-5.

46. Forensic Joint Photographic Experts Group (JPEG) Watermarking for Disk Image Leak Attribution: An Adaptive Discrete Cosine Transform–Discrete Wavelet Transform (DCT-DWT) Approach / B. I. Onyeashie et al. Electronics. 2025. Vol. 14. P. 1800. URL: https://doi.org/10.3390/electronics14091800.

47. Edge-Aware Dual-Task Image Watermarking Against Social Network Noise / H. Jiang et al. Applied Sciences. 2025. Vol. 15. P. 57. URL: https://doi.org/10.3390/app15010057.

48. Bsoul A. A. R. K., Ismaille A. B. Optimizing Image Watermarking with Dual-Tree Complex Wavelet Transform and Particle Swarm Optimization.

49. Resampling-Detection-Network-Based Robust Image Watermarking against Scaling and Cutting / H.-L. Li et al. Sensors. 2023. Vol. 23. P. 8195. URL: https://doi.org/10.3390/s23198195.

50. High Capacity Reversible Data Hiding in Encrypted Images Based on Adaptive Quadtree Partitioning and MSB Prediction / K. Qi, M. Zhang, F. Di, Y. Kong. Frontiers of Information Technology & Electronic Engineering. 2023. Vol. 24, no. 8. P. 1156–1168. URL: https://doi.org/10.1631/FITEE.2200501.

51. A blind and robust color image watermarking scheme based on DCT and DWT domains / A. O. Mohammed et al. Multimedia Tools and Applications. 2023. Vol. 82. P. 32855–32881. URL: https://doi.org/10.1007/s11042-023-14797-0.

52. Forensic Joint Photographic Experts Group (JPEG) Watermarking for Disk Image Leak Attribution: An Adaptive Discrete Cosine Transform–Discrete Wavelet Transform (DCT-DWT) Approach / B. I. Onyeashie et al. Electronics. 2025. Vol. 14. P. 1800. URL: https://doi.org/10.3390/electronics14091800.

53. Robust Blind Image Watermarking Using Coefficient Differences of Medium Frequency between Inter-Blocks / B. Zhu, X. Fan, T. Zhang, X. Zhou. Electronics. 2023. Vol. 12. P. 4117. URL: https://doi.org/10.3390/electronics12194117.

54. Reddy K. T., Reddy S. N. A Novel Blind Double-Color Image Watermarking Algorithm Utilizing Walsh–Hadamard Transform with Symmetric Embedding Locations. Symmetry. 2024. Vol. 16. P. 877. URL: https://doi.org/10.3390/sym16070877.

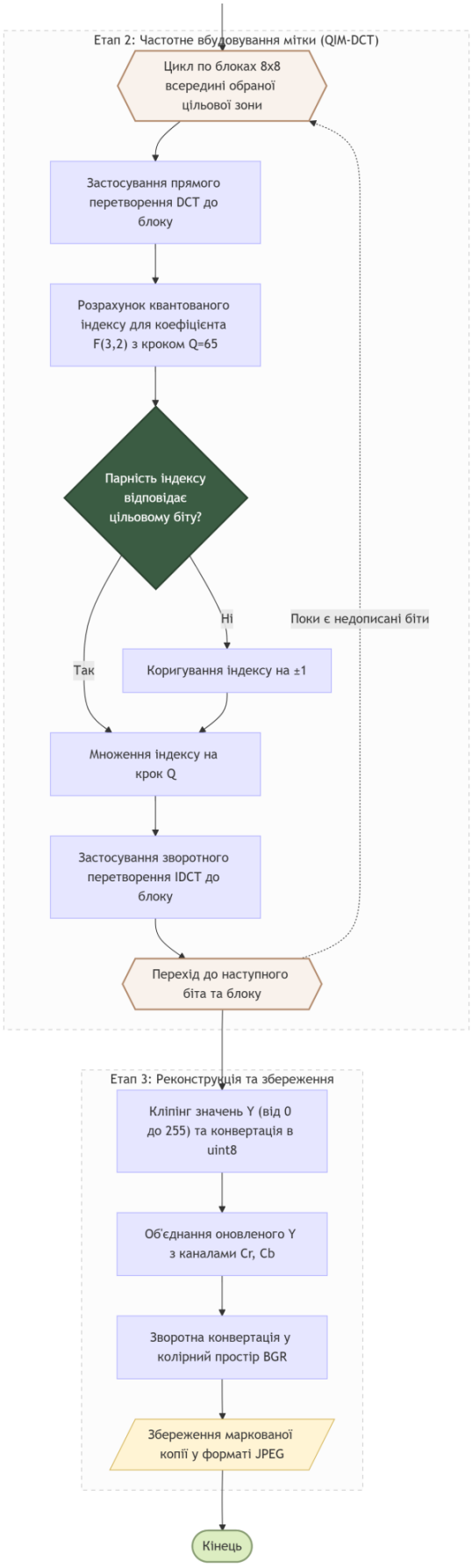
55. Hu H.-T., Hsu L.-Y., Wu S.-T. Blind Watermarking for Hiding Color Images in Color Images with Super-Resolution Enhancement. Sensors. 2023. Vol. 23. P. 370. URL: https://doi.org/10.3390/s23010370.

ДОДАТОК А

Схема алгоритму вбудовування цифрового водяного знака (початок)

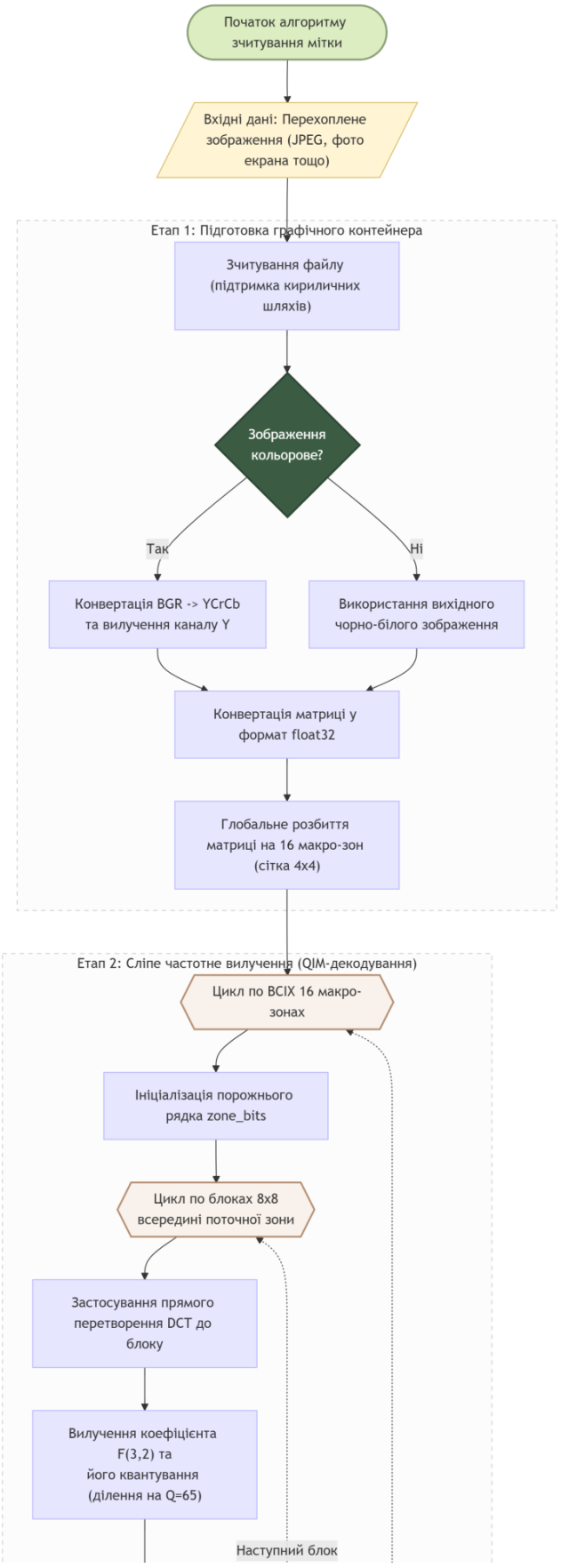
ДОДАТОК Б

Схема алгоритму вбудовування цифрового водяного знака (продовження)



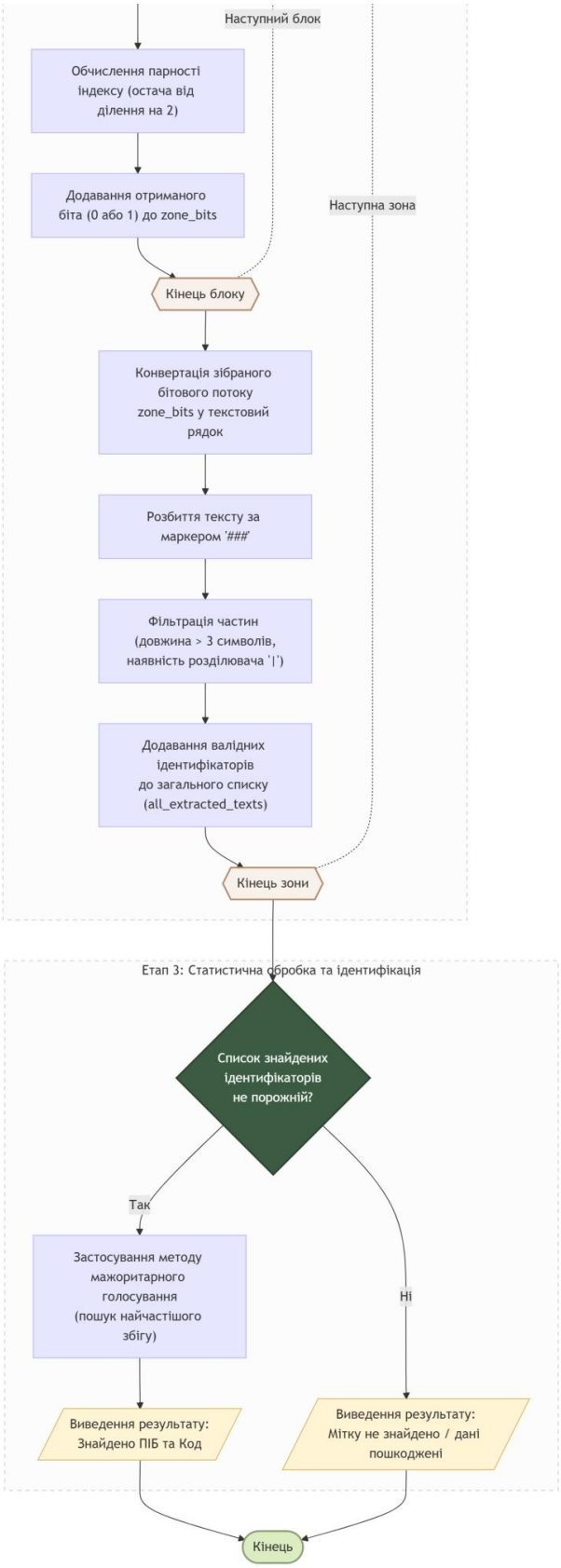
ДОДАТОК В

Схема алгоритму вилучення та декодування мітки (початок)



ДОДАТОК Г

Схема алгоритму вилучення та декодування мітки (продовження)



ДОДАТОК Д

Лістинг програмного коду прототипу системи

```
# -*- coding: utf-8 -*-
import customtkinter as ctk
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import cv2
import numpy as np
import pyodbc
import os
from PIL import Image, ImageTk
import collections

# ============================================================
# НАЛАШТУВАННЯ
# ============================================================
BACKGROUND_IMAGE_PATH = r"C:\Users\onixl\OneDrive\Desktop\bgr1.png"
COLOR_FRAME = "#E6E2CF"
COLOR_ACCENT_1 = "#426546"
COLOR_ACCENT_2 = "#889D73"
COLOR_TEXT_MAIN = "#244539"
HOVER_COLOR = "#2F4F36"

ctk.set_appearance_mode("Light")
ctk.set_default_color_theme("blue")

# ============================================================
# ЛОГІКА (Core) - Content-Aware (Стабільна версія 4x4)
# ============================================================
Q = 65
COEFF_X = 3
COEFF_Y = 2
DELIMITER = "###"
DB_SERVER = 'DESKTOP-O20490K'
DB_NAME = 'SBU_SecureDB'
```

```
CONNECTION_STRING = f'DRIVER={{ODBC Driver 17 for SQL
Server}};SERVER={DB_SERVER};DATABASE={DB_NAME};Trusted_Connection=yes;'


class WatermarkEngine:
    @staticmethod
    def text_to_bits(text):
        return ''.join(format(ord(c), '08b') for c in text)

    @staticmethod
    def bits_to_text(bits):
        chars = []
        for i in range(0, len(bits), 8):
            byte = bits[i:i+8]
            if len(byte) < 8: break
            try:
                char_code = int(byte, 2)
                if 32 <= char_code <= 11000: chars.append(chr(char_code))
                else: chars.append('?')
            except ValueError: chars.append('?')
        return "".join(chars)

    @staticmethod
    def embed(image_path, secret_text, output_path):
        full_text = secret_text + DELIMITER
        base_bitstream = WatermarkEngine.text_to_bits(full_text)

        #підтримка кирилиці
        image_stream = np.fromfile(image_path, dtype=np.uint8)
        image = cv2.imdecode(image_stream, cv2.IMREAD_COLOR)
        if image is None: raise ValueError("Помилка читання файлу. Перевірте
формат або шлях.")

        image_ycc = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)
        Y, Cr, Cb = cv2.split(image_ycc)
        Y = np.float32(Y)
        h, w = Y.shape

        h_blocks, w_blocks = h // 8, w // 8
```

```
        grid_rows, grid_cols = 4, 4
        zone_h = h_blocks // grid_rows
        zone_w = w_blocks // grid_cols

        best_zone = (0, 0)
        max_variance = -1

        for gr in range(grid_rows):
            for gc in range(grid_cols):
                y_start = gr * zone_h * 8
                y_end = y_start + zone_h * 8
                x_start = gc * zone_w * 8
                x_end = x_start + zone_w * 8

                zone_pixels = Y[y_start:y_end, x_start:x_end]
                variance = np.var(zone_pixels)

                if variance > max_variance:
                    max_variance = variance
                    best_zone = (gr, gc)

        start_row = best_zone[0] * zone_h
        end_row = start_row + zone_h
        start_col = best_zone[1] * zone_w
        end_col = start_col + zone_w

        zone_capacity = zone_h * zone_w
        if len(base_bitstream) > zone_capacity:
            raise ValueError(f"Текст занадто довгий для обраної зони (потрібно
{len(base_bitstream)} блоків, є {zone_capacity})!")

        repeats = zone_capacity // len(base_bitstream)
        if repeats == 0: repeats = 1
        bitstream = base_bitstream * repeats

        bit_index = 0
        for i in range(start_row, end_row):
            for j in range(start_col, end_col):
```

```
                if bit_index >= len(bitstream): break

                block = Y[i*8 : (i+1)*8, j*8 : (j+1)*8]
                dct_block = cv2.dct(block)

                coeff = dct_block[COEFF_X, COEFF_Y]
                target_bit = int(bitstream[bit_index])

                quantized = round(coeff / Q)
                if quantized % 2 != target_bit:
                    if coeff >= quantized * Q: quantized += 1
                    else: quantized -= 1
                dct_block[COEFF_X, COEFF_Y] = quantized * Q

                Y[i*8 : (i+1)*8, j*8 : (j+1)*8] = cv2.idct(dct_block)
                bit_index += 1

        Y = np.clip(Y, 0, 255).astype(np.uint8)
        watermarked = cv2.cvtColor(cv2.merge((Y, Cr, Cb)), cv2.COLOR_YCrCb2BGR)

        #підтримка кирилиці при збереженні
        is_success, im_buf_arr = cv2.imencode(".jpg", watermarked,
[int(cv2.IMWRITE_JPEG_QUALITY), 100])
        if is_success:
            im_buf_arr.tofile(output_path)
        return bit_index

    @staticmethod
    def scan(image_path):
        #підтримка кирилиці
        image_stream = np.fromfile(image_path, dtype=np.uint8)
        img = cv2.imdecode(image_stream, cv2.IMREAD_COLOR)
        if img is None: raise ValueError("Помилка читання файлу.")

        if len(img.shape) == 3:
            Y = np.float32(cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)[:, :, 0])
        else: Y = np.float32(img)
        h, w = Y.shape
```

```
        h_blocks, w_blocks = h // 8, w // 8
        grid_rows, grid_cols = 4, 4
        zone_h = h_blocks // grid_rows
        zone_w = w_blocks // grid_cols

        all_extracted_texts = []

        for gr in range(grid_rows):
            for gc in range(grid_cols):
                start_row = gr * zone_h
                end_row = start_row + zone_h
                start_col = gc * zone_w
                end_col = start_col + zone_w

                zone_bits = ""
                for i in range(start_row, end_row):
                    for j in range(start_col, end_col):
                        block = Y[i*8 : (i+1)*8, j*8 : (j+1)*8]
                        quantized = round(cv2.dct(block)[COEFF_X, COEFF_Y] / Q)
                        zone_bits += str(int(quantized % 2))

                raw_text = WatermarkEngine.bits_to_text(zone_bits)
                parts = raw_text.split(DELIMITER)

                valid_parts = [p for p in parts if len(p) > 3 and p.count('|') >=
1]
                all_extracted_texts.extend(valid_parts)

        if all_extracted_texts:
            counter = collections.Counter(all_extracted_texts)
            best_match = counter.most_common(1)[0][0]
            return True, best_match

        return False, "Мітку не знайдено або дані сильно пошкоджені."

class DBManager:
    @staticmethod
```

```
    def get_personnel():
        try:
            conn = pyodbc.connect(CONNECTION_STRING)
            cursor = conn.cursor()
            cursor.execute("SELECT FullName, Department, ServiceCode FROM
Personnel")
            rows = [list(row) for row in cursor.fetchall()]
            conn.close()
            return rows
        except:
            return [['Demo User', 'IT', '001'], ['Test User', 'HR', '002']]

# ============================================================
# GUI APPLICATION
# ============================================================
class PolishedApp(ctk.CTk):
    def __init__(self):
        super().__init__()

        self.title("DLP SECURITY SYSTEM")
        self.geometry("1100x750")

        self._set_background()

        self.grid_columnconfigure(0, weight=3)
        self.grid_columnconfigure(1, weight=2)
        self.grid_rowconfigure(0, weight=1)

        self._init_left_panel()
        self._init_right_panel()

    def _set_background(self):
        try:
            bg_image = Image.open(BACKGROUND_IMAGE_PATH)
            self.bg_photo = ctk.CTkImage(light_image=bg_image, size=(1920, 1080))
            self.bg_label = ctk.CTkLabel(self, text="", image=self.bg_photo)
            self.bg_label.place(x=0, y=0, relwidth=1, relheight=1)
            self.bg_label.lower()
```

```
        except Exception:
            self.configure(fg_color="#F4F0DD")


    def _init_left_panel(self):
        self.frame_gen = ctk.CTkFrame(self, fg_color=COLOR_FRAME,
corner_radius=10, border_width=0, bg_color="transparent")
        self.frame_gen.grid(row=0, column=0, sticky="nsew", padx=20, pady=20)


        lbl_title = ctk.CTkLabel(self.frame_gen, text="МОДУЛЬ МАРКУВАННЯ",
font=("Segoe UI", 16, "bold"), text_color=COLOR_TEXT_MAIN)
        lbl_title.pack(anchor="w", padx=25, pady=(25, 10))


        lbl_step1 = ctk.CTkLabel(self.frame_gen, text="Оригінальний документ:",
font=("Segoe UI", 14), text_color=COLOR_TEXT_MAIN)
        lbl_step1.pack(anchor="w", padx=25, pady=(10, 5))


        self.btn_file_enc = ctk.CTkButton(self.frame_gen, text="Завантажити
зображення", fg_color=COLOR_ACCENT_1, hover_color=HOVER_COLOR, text_color="white",
corner_radius=8, height=40, font=("Segoe UI", 12),
command=self.select_encode_file)
        self.btn_file_enc.pack(fill="x", padx=25, pady=5)


        self.lbl_enc_filename = ctk.CTkLabel(self.frame_gen, text="Файл не
обрано", text_color="gray")
        self.lbl_enc_filename.pack(anchor="w", padx=30)


        ctk.CTkFrame(self.frame_gen, height=2, fg_color="white").pack(fill="x",
padx=25, pady=15)


        lbl_step2 = ctk.CTkLabel(self.frame_gen, text="Отримувачі (Редагування:
подвійний клік по примітці):", font=("Segoe UI", 14), text_color=COLOR_TEXT_MAIN)
        lbl_step2.pack(anchor="w", padx=25, pady=(5, 5))


        table_frame = ctk.CTkFrame(self.frame_gen, fg_color="transparent")
        table_frame.pack(fill="both", expand=True, padx=25, pady=5)


        style = ttk.Style()
        style.theme_use("clam")
        style.configure("Treeview", background="#FDFCF5", foreground="#333",
fieldbackground="#FDFCF5", rowheight=35, font=("Segoe UI", 11), borderwidth=0)
        style.configure("Treeview.Heading", background=COLOR_ACCENT_2,
foreground="white", relief="flat", font=("Segoe UI", 11, "bold"))
```

```
        style.map("Treeview", background=[('selected', COLOR_ACCENT_1)])


        self.tree = ttk.Treeview(table_frame, columns=("Select", "Name", "Dept",
"Code", "Note"), show="headings", selectmode="browse")
        self.tree.heading("Select", text="[X]")
        self.tree.heading("Name", text="ПІБ")
        self.tree.heading("Dept", text="Деп.")
        self.tree.heading("Code", text="Код")
        self.tree.heading("Note", text="Індивідуальна примітка")


        self.tree.column("Select", width=40, anchor="center")
        self.tree.column("Name", width=160)
        self.tree.column("Dept", width=80, anchor="center")
        self.tree.column("Code", width=80, anchor="center")
        self.tree.column("Note", width=180)


        self.tree.pack(side="left", fill="both", expand=True)
        self.tree.bind('<Button-1>', self._on_tree_click)
        self.tree.bind('<Double-1>', self._on_tree_double_click)
        self.load_data()


        self.btn_run = ctk.CTkButton(self.frame_gen, text="ЗГЕНЕРУВАТИ
ПЕРСОНАЛІЗОВАНІ КОПІЇ", fg_color=COLOR_ACCENT_1, hover_color=HOVER_COLOR,
text_color="white", font=("Segoe UI", 13, "bold"), height=50, corner_radius=8,
command=self.run_batch)
        self.btn_run.pack(fill="x", padx=25, pady=25)


    def _init_right_panel(self):
        self.frame_dec = ctk.CTkFrame(self, fg_color=COLOR_FRAME,
corner_radius=10, border_width=0, bg_color="transparent")
        self.frame_dec.grid(row=0, column=1, sticky="nsew", padx=(0, 20), pady=20)


        lbl_title = ctk.CTkLabel(self.frame_dec, text="МОДУЛЬ АНАЛІЗУ",
font=("Segoe UI", 16, "bold"), text_color=COLOR_TEXT_MAIN)
        lbl_title.pack(anchor="w", padx=25, pady=(25, 10))


        lbl_d1 = ctk.CTkLabel(self.frame_dec, text="Перехоплений файл:",
font=("Segoe UI", 14), text_color=COLOR_TEXT_MAIN)
        lbl_d1.pack(anchor="w", padx=25, pady=(10, 5))
```

```
        self.lbl_dec_filename = ctk.CTkLabel(self.frame_dec, text="Файл не
обрано", text_color="gray")
        self.lbl_dec_filename.pack(anchor="w", padx=30)


        self.btn_file_dec = ctk.CTkButton(self.frame_dec, text="Обрати файл",
fg_color=COLOR_ACCENT_1, hover_color=HOVER_COLOR, text_color="white",
corner_radius=8, command=self.select_decode_file)
        self.btn_file_dec.pack(fill="x", padx=25, pady=10)


        ctk.CTkFrame(self.frame_dec, height=2, fg_color="white").pack(fill="x",
padx=25, pady=15)


        lbl_d2 = ctk.CTkLabel(self.frame_dec, text="Результат експертизи:",
font=("Segoe UI", 14), text_color=COLOR_TEXT_MAIN)
        lbl_d2.pack(anchor="w", padx=25, pady=5)


        self.txt_res = ctk.CTkTextbox(self.frame_dec, fg_color="#FDFCF5",
text_color="#333", corner_radius=8, font=("Consolas", 12))
        self.txt_res.pack(fill="both", expand=True, padx=25, pady=10)


        self.btn_scan = ctk.CTkButton(self.frame_dec, text="АНАЛІЗУВАТИ",
fg_color=COLOR_ACCENT_1, hover_color=HOVER_COLOR, text_color="white",
corner_radius=8, height=50, font=("Segoe UI", 13, "bold"),
command=self.run_decode)
        self.btn_scan.pack(fill="x", padx=25, pady=25)


    def load_data(self):
        for i in self.tree.get_children(): self.tree.delete(i)
        data = DBManager.get_personnel()
        for row in data:
            self.tree.insert("", "end", values=['[ ]'] + row + [""])


    def _on_tree_click(self, event):
        if self.tree.identify_column(event.x) == '#1':
            item = self.tree.identify_row(event.y)
            if item:
                vals = list(self.tree.item(item, "values"))
                vals[0] = '[X]' if vals[0] == '[ ]' else '[ ]'
                self.tree.item(item, values=vals)


    def _on_tree_double_click(self, event):
```

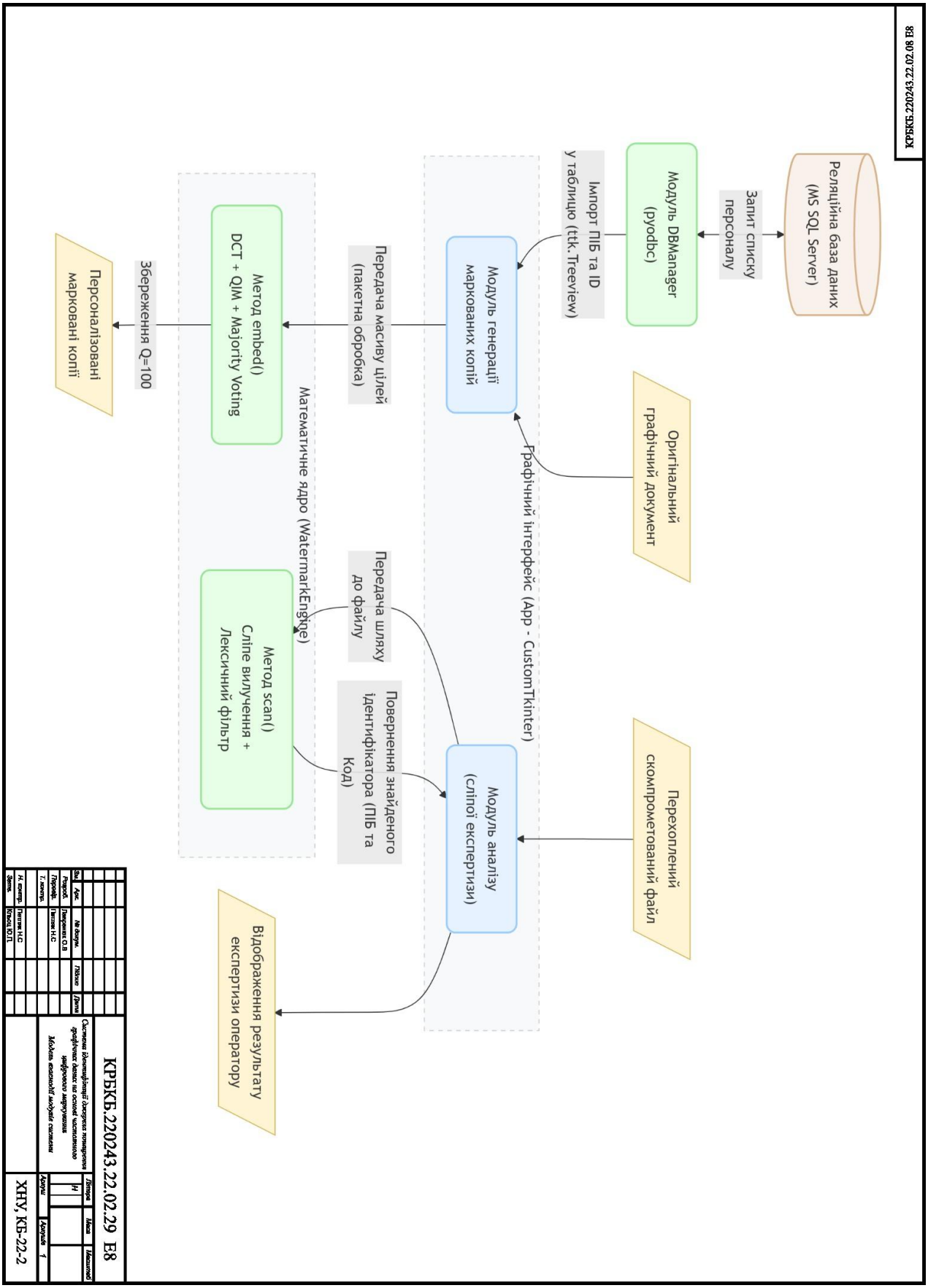
```
        if self.tree.identify_column(event.x) == '#5':
            item = self.tree.identify_row(event.y)
            if not item: return
            x, y, w, h = self.tree.bbox(item, '#5')
            entry = tk.Entry(self.tree, bg="white", fg="black", font=("Segoe UI",
11), relief="solid", borderwidth=1)
            entry.place(x=x, y=y, width=w, height=h)
            entry.insert(0, self.tree.item(item, "values")[4])
            entry.focus()

            def save(e):
                vals = list(self.tree.item(item, "values"))
                vals[4] = entry.get()
                if vals[4].strip(): vals[0] = '[X]'
                self.tree.item(item, values=vals)
                entry.destroy()

            def cancel(e):
                entry.destroy()

            entry.bind('<Return>', save)
            entry.bind('<FocusOut>', save)
            entry.bind('<Escape>', cancel)

    def select_encode_file(self):
        p = filedialog.askopenfilename()
        if p:
            self.encode_path = p
            self.lbl_enc_filename.configure(text=os.path.basename(p),
text_color=COLOR_ACCENT_2)

    def select_decode_file(self):
        p = filedialog.askopenfilename()
        if p:
            self.decode_path = p
            self.lbl_dec_filename.configure(text=os.path.basename(p),
text_color=COLOR_ACCENT_2)
```

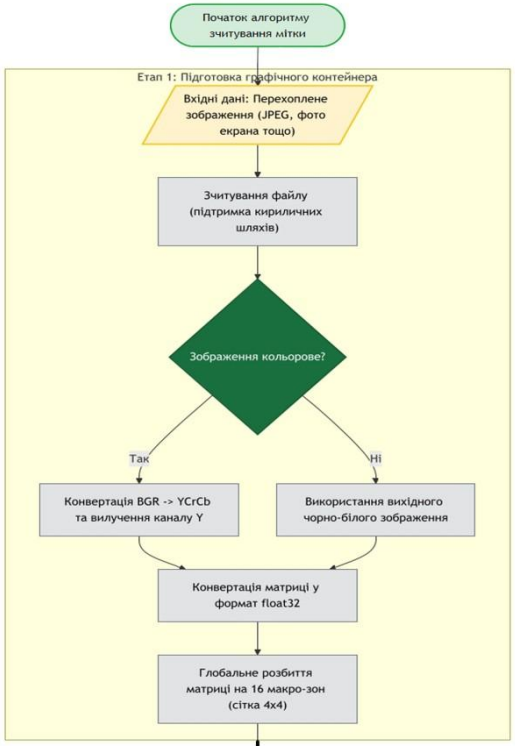
```
    def run_batch(self):
        if not hasattr(self, 'encode_path'): return
        targets = [self.tree.item(i, "values") for i in self.tree.get_children()
if self.tree.item(i, "values")[0] == '[X]']
        if not targets: return

        save_dir = filedialog.askdirectory()
        if not save_dir: return

        cnt = 0
        for t in targets:
            name = str(t[1]).replace("'", "").replace("(", "")
            code = str(t[3]).replace("'", "").replace(")", "").replace(",", "")

            #зменш. об'єм даних
            text = f"{name}|{code}"

            fname =
f"{os.path.splitext(os.path.basename(self.encode_path))[0]}_{code}.jpg"
            try:
                WatermarkEngine.embed(self.encode_path, text,
os.path.join(save_dir, fname))
                cnt += 1
            except Exception as e:
                messagebox.showerror("Помилка генерації", f"Збій на користувачі
{name}!\nПричина: {str(e)}")
                return

        if cnt > 0:
            messagebox.showinfo("Успіх", f"Згенеровано {cnt} маркованих копій.")

    def run_decode(self):
        if not hasattr(self, 'decode_path'): return
        self.txt_res.delete("1.0", "end")

        try:
            found, res = WatermarkEngine.scan(self.decode_path)
            if found:
                p = res.split('|')
```

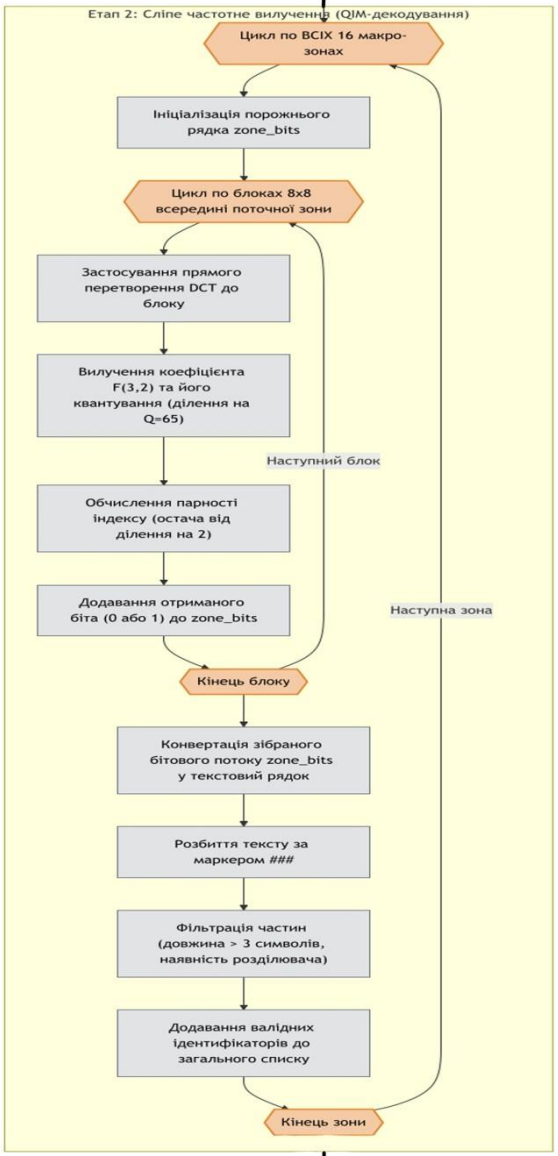
```
                self.txt_res.insert("end", "ВИТІК ВИЯВЛЕНО\n\n")
                if len(p) >= 2:
                    self.txt_res.insert("end", f"ПІБ:  {p[0]}\n")
                    self.txt_res.insert("end", f"Код:  {p[1]}\n")
                else:
                    self.txt_res.insert("end", f"Дані: {res}\n")
            else:
                self.txt_res.insert("end", "□ " + res)
        except Exception as e:
            self.txt_res.insert("end", f"Помилка сканування: {str(e)}")

if __name__ == "__main__":
    app = PolishedApp()
    app.mainloop()
```

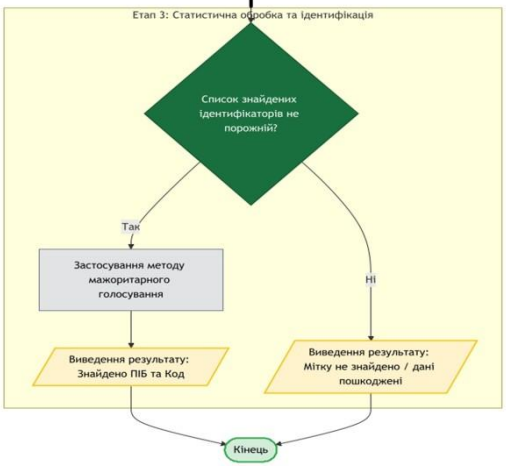
ДОДАТОК Е

Зменшені копії аркушів графічної частини







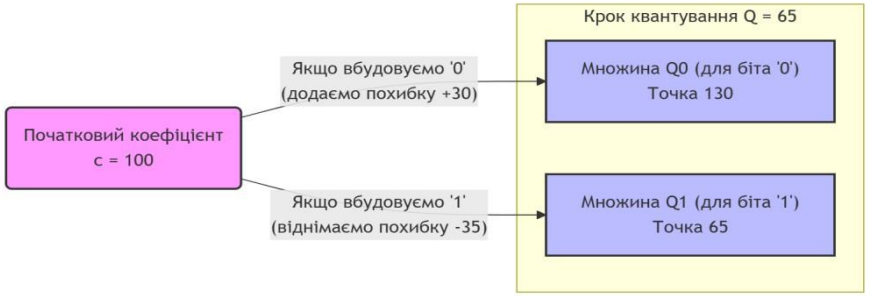


					КРБКБ.220243.22.02.29 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Система ідентифікації джерела поширення графічних даних на основі частотного цифрового маркування Схема алгоритму сліпого вилучення мітки	Літера	Маса / Масштаб
Розроб.		Поварених О.В				Н	
Перевір.		Пилтник Н.С				Аркуш	Аркушів 1
Т. контр.							
Н. контр.		Пилтник Н.С				ХНУ, КБ-22-2	
Затв.		Кльоц Ю.П.					

МАТЕМАТИЧНА МОДЕЛЬ АДАПТИВНОГО ЧАСТОТНОГО МАРКУВАННЯ

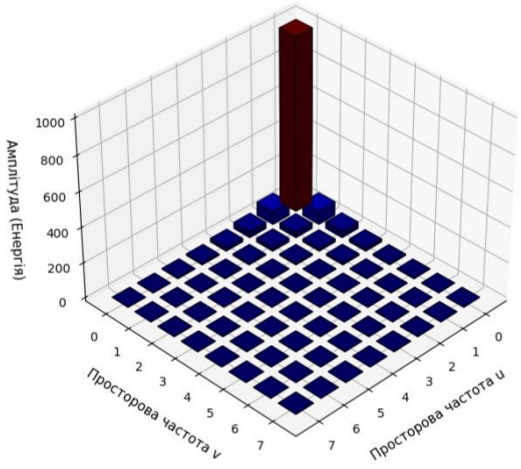
Декомпозиція колірного простору (YCrCb)

Математика вбудовування



Частотне перетворення

Ефект ущільнення енергії (Energy Compaction) у блоці DCT 8x8



Захист від втрат

Механізм формування просторової надмірності (Spatial Redundancy)

					КРБКБ.220243.22.02.29 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Система ідентифікації джерела походження графічних даних на основі частотного цифрового маркування. Математична модель частотного маркування	Літера	Маса	Масштаб
Розроб.		Павренюк О.В				Н		
Перевір.		Пятник Н.С						
Т. контр.						Аркуш	Аркушів 1	
Н. контр.		Пятник Н.С				ХНУ, КБ-22-2		
Затв.		Кльоц Ю.П.						