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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

"Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3 "

Виконала: здобувач 2-го курсу, групи КІТС-24м
спеціальності 125-Кібербезпека та захист інформації
Освітня програма - Кібербезпека інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Прокопович-Гузенко Л. В.
(прізвище та ініціали)

Керівник: к.т.н., доц., доцент каф. МБІС
Карпінець В. В.
(прізвище та ініціали)
«09» 12 2025 р.

Опонент: к.т.н., доц., доц. каф. ОТ
Крупельницький Л. В.
(прізвище та ініціали)
«04» 12 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

Юрій ЯРЕМЧУК
"09" 12 2025 р.

Вінниця ВНТУ - 2025 рік

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

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

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС
_____ Юрій ЯРЕМЧУК
"___" _____ 2025р.

ЗАВДАННЯ
на магістерський кваліфікаційну роботу студенту
Прокопович-Гузенко Лідія Володимирівна .
(прізвище, ім'я, по-батькові)

1. Тема роботи
2. Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3

Керівник роботи к.т.н., доц., доцент каф. МБІС Карпінець Василь Васильович
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "24" вересня 2025 року №313.

2. Строк подання студентом роботи за тиждень до захисту.

3. Вихідні дані роботи: Стандарти, електронні джерела, підручники та наукові статті по темі, які стосуються теми магістерської кваліфікаційної роботи.

4. Зміст текстової частини: Для досягнення поставленої мети було визначено такі основні завдання: дослідити сучасні методи цифрового водяного знакування та їхню стійкість до різних типів атак; виконати аналіз переваг і недоліків наявних підходів; розробити метод із адаптивним вибором радіуса ключа; реалізувати та протестувати запропонований алгоритм. У першому розділі проведено аналіз принципів роботи DCT та DWT у водяному знакуванні, а також оцінено їхню стійкість до атак. У другому розділі здійснено розроблення алгоритмів вбудовування та екстракції цифрового водяного знака з використанням адаптивного вибору радіуса ключа. У третьому розділі виконано практичну реалізацію і тестування запропонованого методу в середовищі Python, а також проведено порівняння з існуючим аналогом, що підтвердило його ефективність.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
В першому розділі магістерської роботи 1 рисунок, в другому розділі 2 рисунків, в третьому розділі 8 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Карпінець В.В. к.т.н., доц., доцент каф.МБІС		

I	Карпінець В.В. к.т.н., доц., доцент каф.МБІС		
II	Карпінець В.В. к.т.н., доц., доцент каф.МБІС		
III	Карпінець В.В. к.т.н., доц., доцент каф.МБІС		
Економічна частина			
IV	Ратушняк О.Г., к.т.н., доцент кафедри ЕПВМ		

7. Дата видачі завдання 24 вересня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи	20.09.2025	29.09.2025	
2.	Аналіз предметної області обраної теми	30.09.2025	12.10.2025	
3.	Розробка роботи	13.10.2025	29.10.2025	
4.	Написання магістерської роботи на основі розробленої теми	30.10.2025	23.11.2025	
5.	Передзахист магістерської роботи	24.11.2025	28.11.2025	
6.	Виправлення, уточнення, коригування магістерської кваліфікаційної роботи	29.11.2025	06.12.2025	
7.	Захист магістерської кваліфікаційної роботи	08.12.2025	12.12.2025	

Студент Прокопович-Гузенко Л.В.
(підпис)

Керівник роботи Карпінець В.В.
(підпис)

АНОТАЦІЯ

УДК:004.056

Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3. Магістерська кваліфікаційна робота зі спеціальності 125 - “Кібербезпека”, освітня програма “Кібербезпека інформаційних технологій та систем”. Вінниця: ВНТУ, 2025.

Укр. мовою. Бібліогр.: 43 назв; рис.: 11; табл.: 15.

У магістерській кваліфікаційній роботі розглянуто проблему підвищення методів цифрового водяного знакування для забезпечення захисту цифрового контенту від несанкціонованого використання. Увага приділяється розробці підвищення стійкості цифрових водяних знаків на основі дискретного косинусного перетворення (DCT), дискретного вейвлет перетворення (DWT), нейромережі Inception V3 та SIFT. Такий підхід дозволяє підвищити стійкість водяних знаків до атак (геометричних, частотних та комбінованих), зберігаючи високу якість зображення.

У роботі проведено аналіз сучасних технологій водяного знакування, переваг та обмежень. Запропоновано новий алгоритм вбудовування цифрового водяного знака. Розроблено алгоритм екстракції водяного знака для відновлення після атак.

Для оцінювання результативності розробленого методу застосовано розширений набір показників якості, до якого увійшли SSIM, PSNR, NCC та BER. Додатково виконано порівняльний аналіз роботи запропонованого алгоритму з наявними підходами, що дозволило комплексно визначити його переваги та недоліки. Усі етапи моделювання та реалізації були виконані в середовищі Google Colab мовою Python, яке забезпечує високу точність

чисельних обчислень, а також дозволяє ефективно поєднувати різні компоненти частотного та просторового аналізу в межах єдиної програмної платформи.

Отримані експериментальні дані переконливо демонструють дієвість запропонованого методу та його здатність надійно зберігати вбудований водяний знак навіть за умов інтенсивних та комплексних атак. Це свідчить про високу робастність алгоритму й підтверджує його придатність для реального використання. Запропонований підхід може бути ефективно впроваджений у практичні системи захисту цифрового контенту в різних сферах, включно з мультимедійними технологіями, медичною візуалізацією та охороною інтелектуальної власності, забезпечуючи надійну автентифікацію та контроль цілісності цифрових матеріалів.

Ключові слова: дискретне вейвлет-перетворення, дискретне косинусне перетворення, нейромережа, DCT, DWT, SIFT, Inception V3, водяний знак.

ANNOTATION

UDC: 004.056

Enhancing the robustness of digital watermarks in the frequency domain of images against geometric attacks based on the SIFT method and the Inception V3 neural network. Master's qualification thesis in specialty 125 – Cybersecurity, educational program Cybersecurity of Information Technologies and Systems. Vinnytsia: VNTU, 2025.

In Ukrainian. Bibliography: 43 sources; figures: 11; tables: 12.

This master's thesis investigates the problem of improving digital watermarking methods to protect digital content from unauthorized use. The focus is placed on increasing the robustness of watermarking by integrating discrete cosine transform (DCT), discrete wavelet transform (DWT), the Inception V3 neural network, and SIFT. This approach enhances watermark resistance to attacks (geometric, frequency-based, and combined) while maintaining high visual quality of the image.

The thesis presents an analysis of existing watermarking technologies, outlining their advantages and limitations. A new watermark embedding algorithm has been proposed, along with an extraction algorithm designed to restore the watermark after various attacks.

To evaluate the performance of the developed method, an extended set of quality metrics was applied, including SSIM, PSNR, NCC, and BER. Additionally, a comparative analysis with existing approaches was conducted, allowing for a comprehensive assessment of strengths and weaknesses. All modelling and implementation stages were carried out in Google Colab using Python, which ensures high numerical precision and enables seamless integration of spatial and frequency-domain components within a unified software environment.

The experimental results clearly demonstrate the effectiveness of the proposed method and its ability to reliably preserve the embedded watermark even under intensive and combined attacks. This confirms the robustness of the algorithm and its

suitability for practical deployment. The proposed approach can be effectively used in digital content protection systems across various domains, including multimedia technologies, medical imaging, and intellectual property protection, providing reliable authentication and integrity control of digital assets.

Keywords: discrete wavelet transform, discrete cosine transform, neural network, DCT, DWT, SIFT, Inception V3, watermark.

ВСТУП

Актуальність. Стрімкий розвиток цифрових технологій, збільшення обсягів обміну мультимедійними даними та глобальна доступність інформації у мережі Інтернет зумовлюють гостру необхідність захисту цифрового контенту від несанкціонованого використання, копіювання та підробки. Одним із найбільш ефективних механізмів забезпечення автентичності та цілісності даних є цифрове водяне знакування, проте традиційні методи вбудовування інформації у зображення часто виявляються вразливими до геометричних атак, таких як поворот, масштабування, обрізання чи зміщення. Це призводить до втрати синхронізації між вбудованим знаком та носієм, що здатне унеможливити відновлення або підтвердження прав власності. Саме тому важливим завданням сучасної наукової спільноти є підвищення стійкості цифрових водяних знаків, зокрема в частотному домені, оскільки він забезпечує кращий баланс між непомітністю, якістю відображення та захищеністю до обробки й стиснення.

У даній магістерській роботі розглядатимемо сучасні методи цифрового водяного знакування, що спрямовані на на захист авторського права та цілісності цифрових зображень. Основний акцент зроблено на підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3.

У ході дослідження розглянуті теоретичні основи сучасних методів цифрового водяного знакування та визначено перспективні підходи для реалізації завдань. Розробка алгоритму вбудування базуватиметься на дискретному косинусному-перетворенню, дискретному вейвлет-перетворенню, картою важливості за допомогою мережі Inception V3 та виявленню ключових точок за допомогою SIFT, що буде підвищувати стійкість цифрових водяних знаків.

Особливу уваги приділятиметься перевіркою стійкості цифрових водяних знаків до різних видів атак таких, як масштабування, стиснення, обертання, шумові впливи. Реалізація алгоритму вбудовування та екстракції водного знаку виконана у середовищі Google Colab, що забезпечить доцільне використання вбудованих бібліотек. Також обчислимо метрики PSNR, SSIM, BER та NCC для розуміння ефективності алгоритму вбудування водяного знаку для атакованного зображення.

Розроблений підхід порівняємо з існуючим методами, де буде показано дієвість алгоритму до геометричних атак.

Мета і задача дослідження. Метою роботи є підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3.

Задачами дослідження є:

- аналіз сучасних методів цифрового водяного знакування, їх переваг, обмежень та стійкості до геометричних і частотних атак.
- розробити алгоритм вбудовування цифрового водяного знака у частотний простір зображення із використанням інваріантних ознак та нейромережевої оцінки областей.
- реалізувати алгоритм екстракції водяного знака з можливістю відновлення після геометричних та комбінованих атак.
- провести експериментальне дослідження ефективності розробленого методу, здійснити тестування на різних типах атак.
- виконати оцінювання результатів за метриками PSNR, SSIM, NCC, BER та порівняти метод з існуючими аналогами.
- здійснити узагальнення результатів, сформувати висновки та рекомендації щодо практичного застосування алгоритму.

Об’єкт дослідження: методи цифрового водяного знакування у частотному просторі.

Предмет дослідження: процес підвищення стійкості алгоритму вбудовування та екстракції водяного знакування.

Новизна роботи: розроблений підхід підвищення стійкості водяного знаку у частотному просторі за рахунок нейромережі Inception V3 та алгоритмом SIFT, що відкриває можливості для створення адаптивних систем водяного знакування, здатних ефективно протистояти як частотним, так і геометричним спотворенням. Застосування інваріантних ключових ознак та глибоких нейромереж дозволяє не лише підвищити стійкість до атак, а й значно покращити процес відновлення водяного знаку після пошкоджень.

Практична цінність: реалізовано алгоритм вбудовування та екстракції ЦВЗ із використанням DCT та DWT, алгоритмом SIFT і нейромережею Inception V3, який показав високу результативність стійкості до частотних та геометричних атак.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ ЦИФРОВОГО ВОДЯНОГО ЗНАКУВАННЯ ДО ГЕОМЕТРИЧНИХ АТАК

Даний розділ містить огляд існуючих підходів до цифрового водяного знакування, де здійснено систематизацію теоретичних основ і практичних рішень, що лежать в основі побудови алгоритмів водяного знакування у просторовому та частотному доменах, а також визначено переваги, недоліки й тенденції розвитку таких методів.

Метою даного розділу є аналіз теоретичних основ та сучасних методів цифрового водяного знакування, оцінка їхніх переваг і недоліків у контексті протидії геометричним атакам.

1.1 Основні особливості цифрового водяного знакування.

Розвиток інформаційних технологій в сучасності сприяв відкриттю нових можливостей зростання обсягів та розповсюдження цифрового контенту, що ставить питання щодо безпеки цілісності та захисту авторських прав, тому зросла необхідність у розробці нових методів безпеки. Одним із таких методів є цифрове водяне знакування, що дозволяє вносити приховану інформацію у цифрові зображення та забезпечує захист від викривлень та підробок.

Цифрове водяне знакування (ЦВЗ) є одним із найефективніших підходів до захисту авторських прав та забезпечення цілісності цифрових зображень. ЦВЗ поділяється на видиме та невидиме залежно від того, чи може користувач візуально розпізнати наявність знаку у зображенні. Видимимі водяні знаки представлені у вигляді логотипів, текстів або символів, які накладають поверх зображення. Невидимі водяні знаки вбудовуються в піксельну або частотну структуру зображення для прихованої автентифікації, відстеження або перевірки власності, щоб не впливати на його визуальне сприйняття та естетику

контенту. Різниця між видимим та невидимим цифровим знакуванням наведено в таблиці 1.

Критерій	Видиме водяне знакування	Невидиме водяне знакування
Видимість для користувача	Помітне для ока, у вигляді логотипу,тексту або символу	Непомітне, вбудоване у пікселі або в частотному просторі
Основна мета	Публічний захист авторських прав, ідентифікація власника	Прихована автентифікація, виявлення підробок
Сфера застосування	Зображення у відкритому доступі (мережі, портфоліо, тощо)	Медичні, правові, військові зображення
Методи реалізації	Просторове накладання	Методи DCT, DWT, SVD, LSB, нейромережеві моделі
Сприйняття користувачем	Може вплинути на естетику та якість зображення	Не змінює візуальну якість контенту
Рівень безпеки	Низький, легко видалити або замаскувати	Високий, складно видалити без втрати якості або руйнування зображення
Стійкість до атак	Низька, вразливе до розмиття, обрізання, зміни кольору, тощо	Висока, стійке до геометричних атак, стиснення, фільтрації
Переваги	Простота в реалізації, швидке сприйняття, підвищення впізнаваності автора	Висока стійкість, непомітність, можливість багаторівневого вбудування

Продовження таблиці 1.

Недоліки	Легко видаляється, втрата естетики зображення	Складність реалізації, потребує спеціального алгоритму для виявлення

Таблиця 1 - порівняння видимого та невидимого водяного знакування.

Невидимі водяні знаки є більш технологічно складними, але забезпечують вищий рівень захисту. Його суть полягає у вбудовування водяного знаку у пікселі або в його коефіцієнти, отримані після перетворення зображення у частотну область. Такий підхід значно ускладнює та підвищує стійкість до атак стиснення, фільтрації, масштабування, повороту та інших спотворень, які найчастіше зустрічаються при обробці мультимедійних даних, застосовують у системах цифрової судової експертизи, автентифікації медичних зображень, захисті даних у хмарних сервісах та цифрових архівах.

Ефективність системи цифрового водяного знакування оцінюється за чотирма ключовими критеріями: непомітність, робастність, ємність та безпека. Ці показники взаємопов’язані та між ними існує компроміс, де покращення одного параметра часто призводить до погіршення іншого.

Параметр непомітності один із ключових характеристик, який визначає наскільки сильно вбудований знак впливає на якість вихідного зображення. Основна мета полягає в тому, щоб після процесу вбудовування візуальні властивості зображення залишалися незмінними для людського ока, тобто щоб знак не спотворював сприйняття кольорів, контрасту, текстур чи деталей. Ідеальна система водяного знакування має забезпечувати високу прозорість при якій різниця між оригінальним та водяним зображенням практично непомітна навіть при детальному порівнянні. Цей параметр оцінюють за допомогою психовізуальних метрик, як пікове відношення сигналу та шуму (PSNR, Peak

Signal-to-Noise Ratio) та індекс структурної подібності (SSIM, Structural Similarity Index). Пікове відношення сигналу та шуму - це відношення між максимально можливою потужністю сигналу та потужністю шуму, спричиненого вбудовуванням знаку. Індекс структурної подібності - це показник структурної подібності між оригінальним і зміненим зображенням.

Непомітність знаходиться у прямому конфлікті з робастністю. Щоб зробити водяний знак більш стійким до атак, необхідно збільшити амплітуду модифікацій у зображенні, що знижує прозорість. Таким чином висока непомітність зазвичай досягається за рахунок зниження стійкості.

Вагомою характеристикою ЦВЗ є робастність, яка характеризує здатність водяного знаку зберігатись після застосування до об’єкта різних спотворень чи атак. Висока стійкість особливо важлива для авторського захисту або аутентифікації, де водяний знак має бути виявлений після обробки.

Для оцінки робасності застосовують набір стандартних атак, які імітують звичайні процеси редагування та навмисні спроби видалення, використовують коефіцієнти кореляції між вихідними та витягнутим знаком та аналіз поведінки після атак. До таких атак відносять стиснення, додавання шуму, фільтрація, зміна масштабу та обертання, обрізання, перетворення кольорів, комбіновані атаки.

Для визначення кількості інформації, яку можна вбудувати у носій без погіршення його якості використовують ємність. Для зображень це може бути кількість бітів або символів, що вбудовуються у пікселі або блоки. Висока ємність дозволяє, наприклад, вставляти метадані, цифрові підписи або ідентифікатори прав власності.

Найважливіша характеристика ЦВЗ - це безпека, яка означає захищеність схеми водяного знакування від несанкціонованого доступу або підробки. Для покращення безпеки використовують секретний ключ для генерації або вбудовування знаку, характеризується криптографічною стійкістю та

невідтворюваністю. Безпека визначає, наскільки система здатна протистояти злому, підробці або несанкціонованому копіюванню.

1.2 Сучасні методи цифрового водяного знакування

Цифрове водяне знакування (ЦВЗ) є одним із ключових механізмів захисту авторських прав та забезпечення автентичності мультимедійного контенту. Його основна мета - вбудувати у цифрове зображення, відео або аудіо приховану інформацію. Сучасні методи цифрового водяного знакування класифікують за такими критеріями: область вбудування (просторове або частотне), спосіб детектування (сліпе або несліпе), стійкість до атак.

Цифровий водяний знак може бути реалізований у різних доменах представлення зображення - просторовому або частотному. Вибір області визначає не лише якість вбудування, а й рівень стійкості атак. В просторовому домені водяний знак вбудовується у піксельні значення зображення, зазвичай у найменш значущі біти, що є простим у реалізації, має низьке обчислювальне навантаження, але низьку стійкість до стискання або обробки зображення. Наприклад, вбудовування у молодші біти яскравості пікселів без помітної зміни кольорів зображення. В частотному домені зображення перетворюється математично через такі методи, як дискретно косинусне перетворення DCT, дискретно вейвлет-перетворення DWT або сингулярного розкладу SVD. Водяний знак вбудовується у певні частотні коефіцієнти, що описують структуру або текстуру зображення. Такий підхід забезпечує стійкість до втрат якості, геометричних трансформацій, стиснення та шуму, оскільки зміни менш помітні для людського зору, але добре зберігаються під час обробки зображення. Наприклад, вбудування водяного знаку у середньочастотні коефіцієнти DCT, щоб уникнути спотворення візуальних деталей, але збереження стійкості до стиснення.

Сучасні методи також включають комбіновані підходи, що дозволяють досягти кращої стійкості до геометричних спотворень та частотних атак, наприклад, DWT та DCT, DWT та SVD, тощо.

Наступний критерій сучасних методів є сліпа та несліпа детекція водяних знаків. Сліпі методи не потребують наявності оригінального зображення для перевірки, що спрощує процес автентифікації та робить такі системи більш практичними у застосуванні. Несліпі методи, навпаки, потребують доступу до початкового (оригінального) зображення, проте забезпечують вище точність і надійність під час ідентифікації вбудованого водяного знаку.

У новітніх дослідженнях спостерігається тенденція до інтеграції традиційних частотних алгоритмів із методами штучного інтелекту та машинного навчання. Це дозволяє автоматично адаптувати місце і силу вбудовування водяного знаку під особливості конкретного зображення. Сучасні науковці зосереджуються на інтеграції алгоритмів глибокого навчання у процес формування водяного знаку. Використання згорткових нейронних мереж та архітектур дають змогу створювати адаптивні схеми вбудування, які враховують локальні особливості зображення. Такі системи динамічно змінюють параметри сили, положення та частотної області вбудування залежно від контексту зображення, що значно підвищує складність для автоматизованого розпізнавання водяних знаків. Завдяки цьому формуються інтелектуальні, контекстно-залежні схеми водяного знакування, стійкі до аналізу з боку моделей машинного навчання. Вони забезпечують не лише збереження високої якості візуального контенту, але й створюють додатковий рівень криптографічної та семантичної захищеності, роблячи процес виявлення або знищення водяного знаку практично неможливим навіть за використання сучасних інструментів штучного інтелекту.

Сучасні підходи до цифрового водяного знакування перебувають у стані активної еволюції, орієнтованої на підвищення стійкості та адаптивності систем захисту зображень. Розвиток технологій спрямований на вдосконалення

способів вбудовування інформації у різні просторові та частотні домени, що дозволяє створювати водяні знаки з покращеними характеристиками надійності та прихованості.

Одним із напрямів є багатошарове або багаторівневе водяне знакування, за якого інформація вбудовується одночасно у декілька діапазонів частот або рівнів розкладу зображення. Такий підхід забезпечує резервування інформації, навіть у разі часткового спотворення зображення водяний знак може бути успішно відновлений із решти рівнів. Поєднання низько- та високочастотних компонентів дозволяє підвищити стійкість системи до фільтрації, додавання шумів, поворотів, масштабування та інших геометричних атак. У результаті багатошарові схеми створюють більш надійні та гнучкі механізми захисту цифрових зображень.

Паралельно з цим активно розвивається напрям адаптивного водяного знакування, який базується на інтелектуальному аналізі структури зображення. Такі алгоритми автоматично визначають найбільш підходящі області для вбудовування з урахуванням текстури, контрасту, градієнтів або зон інтересу (ROI). Завдяки цьому водяний знак інтегрується у ті частини зображення, де його наявність є візуально непомітною, але водночас максимально захищеною від видалення чи спотворення.

Ще одним перспективним напрямом розвитку ЦВЗ є використання квантових принципів для підвищення рівня безпеки та унеможливлення несанкціонованого втручання. Квантові методи водяного знакування базуються на особливостях квантової інформації, зокрема на суперпозиціх станів та принципі невимірюваності, що унеможливлює спостереження або копіювання даних без їхнього спотворення. Це дозволяє створювати водяні знаки, які реагують на будь-яку спробу втручання, оскільки навіть мінімальний зовнішний вплив змінює квантовий стан закодованої інформації.

Сучасні методи ЦВЗ демонструють стрімкий розвиток у напрямі підвищення надійності, гнучкості та інтелектуальної адаптації систем захисту

цифрового контенту. Еволюція від простих методів у піксельному домені до складних гібридних алгоритмів на основі DCT, DWT, SVD та нейронних мереж свідчить про прагнення забезпечити баланс між непомітністю, стійкістю та ємністю вбудування.

1.3 Огляд методів на основі дискретного косинусного перетворення (DCT) та дискретного вейвлет перетворення (DWT).

Методи цифрового знакування у частотному просторі базуються на концепції перетворення зображення у форму, яка розкриває його частотні характеристики. Серед цих підходів, що найчастіше використовуються , є дискретне косинусне перетворення (DCT) та дискретне вейвлет перетворення (DWT), що пропонують компроміс між нечутністю водяного знака та стійкістю атак.

Дискретне косинусне перетворення є одним із найстаріших і водночас найефективніших інструментів частотного аналізу, який широко використовується в алгоритмах стиснення JPEG. Принцип роботи DCT полягає у представленні зображення як суми косинусних функцій різних частот. Це дозволяє розділити інформацію на низькочастотні, середньочастотні та високочастотні компоненти, що характеризують відповідно загальну структуру, деталі та шуми зображення.

У математичній формі DCT для двовимірного сигналу розміром N*M визначається як:

$$C(u,v) = \frac{2}{\sqrt{MN}} * \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) cos\left[\frac{(2x+1)u\pi}{2M}\right] cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

де $f(x,y)$ - інтенсивність пікселя, а $C(u,v)$ - коефіцієнт DCT для частот u, v ; $\alpha(u)\alpha(v)$ - нормувальні коефіцієнти, що дорівнюють $\frac{1}{\sqrt{2}}$ при u, v = 0 і 1 в інших випадках.

Спочатку проводиться попередня обробка, зображення переводиться у відтінки сірого або розділяється на колірні канали (R, G, B), вибирається канал або матриця яскравості Y для вбудовування, оскільки людське око чутливе до змін яскравості. У більшості алгоритмів DCT-водяного знакування зображення ділиться на блоки 8*8 або 16*16 пікселів, такі розміри збалансовують якість та швидкість обчислень. Наступний крок, для кожного блока обчислюється DCT, отримуючи матрицю коефіцієнтів, де елемент (0,0) - це середня яскравість блока (DC-компонент), інші - деталі різних частот (AC-компоненти). Саме вбудування водяного знаку починається з вибору підмножина середньочастотних коефіцієнтів, в які впроваджуються інформація за певним правилом. Наприклад, додавання псевдовипадкової послідовності до вибраних коефіцієнтів, згенерованої ключем або адаптивне масштабування коефіцієнтів, що враховує локальну енергію блока. Після модифікації коефіцієнтів виконується обернене DCT, блоки об’єднуються у фінальне зображення з водяним знаком.

У процесі ЦВЗ в частотному простоі важливим є не лише вибір області, але й спосіб модифікації коефіцієнтів DCT. Залежно від способу зміни частотних коефіцієнтів підходи поділяються на адитивні, мультиплікативні, реляційні та гібридні.

Адитивний підхід найпростіший та найдавніший метод вбудування водяного знаку у DCT-домені. Його суть полягає в додаванні матриці водяного знаку до вибраних коефіцієнтів DCT згідно з формулою:

$$C'(u,v) = C(u,v) + k * W(u,v),$$

де $C(u,v)$ - початковий DCT-коефіцієнт, $W(u,v)$ - елемент матриці водяного знаку, k - коефіцієнт масштабування, який визначає інтенсивність вбудування, де k вибирається емпірично, тобто надто велике значення робить знак помітним, а мале - ускладнює його виявлення.

Мультиплікативне водяне знакування враховує величину самого коефіцієнта DCT, змінюючи його пропорційно, що робить алгоритм стійкішим до атак та має такий вигляд:

$$C'(u,v) = C(u,v) + (1 + k * W(u,v)).$$

Цей підхід вважається енергетично залежним, адже внесена зміна масштабується відносно величини коефіцієнта. Тобто, чим більший коефіцієнт, тим сильніше він модифікується. Це підвищує стійкість, оскільки слабкі компоненти змінюються мінімально, а важливіші - більш виражено.

Реляційне або відносне водяне знакування засноване на зміні співвідношень між кількома коефіцієнтами DCT. Замість модифікації абсолютних значень, змінюється порядок чи співвідношення амплітуд між вибраними частотами. Наприклад, якщо два коефіцієнти $C1$ та $C2$ використовуються для кодування біта водяного знаку, якщо біт =1, то $|C1| > |C2|$, але якщо біт = 0, тоді $|C1| < |C2|$.

Гібридні методи у сфері сучасних досліджень цифрового водяного знакування показують, що найкращі результати забезпечують гібридні методи, які поєднують DCT із іншими перетвореннями. Наприклад, DCT+SVD, у цій моделі після застосування DCT до блоків зображення виконується сингулярний розклад, який дає змогу вбудовувати знак у сингулярні значення, що є стабільними навіть після обробки зображення та має такий вигляд:

$$A = USV^T,\ S` = S + k * W,\ A` = US`V^T.$$

DCT+DWT підхід виконує DCT лише для вибраних піддіапазонів DWT (HL або LH). Завдяки багаторівневості DWT і енергетичній компактності DCT досягається висока непомітність та надійність.

Для кращого розуміння різницю переваг та недоліків розглянемо порівняльну таблицю типів DCT-методів:

Тип методу	Основна ідея	Переваги	Недоліки	Рівень стійкості

Адитивний	Додавання водяного знаку до коефіцієнтів	Простота, швидкість	Низька стійкість до шумів	Низький

Продовження таблиці 2.

Мультиплікативний	Масштабування коефіцієнтів пропорційно їх величині	Вища стійкість, адаптивність	Складніше відновлення	Середній
Реляційний	Зміна співвідношення між коефіцієнтами	Висока стійкість, непомітність	Мала ємність, вразливість до обертання	Високий
Гібридний	Комбінація DCT з іншими перетвореннями	Максимальна стійкість, гнучкість	Висока обчислювальна складність	Дуже високий

Таблиця 2 - порівняльна таблиця типів DCT- методів

Завдяки роботі в частотному просторі ці методи стійкі до JPEG-стиснення, фільтрації, додавання шумів, зміни контрасту, яскравості, обрізання зображення. Однак вони чутливі до геометрични атак - поворотів, масштабування та зсувів, які змінюють просторову структуру блоків і порушують їхню узгодженість. Тому в сучасних дослідженнях для компенсації цього недоліку використовують SIFT (Scale-Invariant Feature Transform), DCT із блоками вирівняними за SIFT-координатами або DCT у поєднанні з вейвлетами для стабілізації просторових характеристик.

Завдяки високої ефективності ці методи використовуються у захисті фотографій та графіки у відкритих мережах, ідентифікації власника авторських

прав у цифрових бібліотеках, контролю цілісності документів та картографічних даних, аудіо та відео водяного знакування у форматах MPEG, AVI, тощо.

Багато алгоритмів цифрового водяного знаку, що базуються на дискретному косинусному перетворенні (ДКП), забезпечують дуже оптимальні компроміси між стійкістю водяного знаку та його невидимістю. Дійсно, ДКП втілює енергетичну компактність та обчислювальну простоту разом із високою стійкістю до частотних спотворень. Тому він може бути показаний для застосувань у захисті авторських прав, а також у системах цифрової автентифікації.

Дискретне вейвлет-перетворення (DWT) є одним із найефективніших інструментів частотного аналізу цифрових зображень, який поєднує частотне та просторове представлення сигналу. На відміну від дискретного косинусного перетворення (DCT), що виконує глобальний аналіз, DWT забезпечує локалізовану декомпозицію, тобто дозволяє аналізувати як дрібні деталі, так і загальну структуру зображення на різних масштабах. Ця властивість робить DWT особливо придатним для цифрового водяного знакування, оскільки забезпечує високу стійкість до атак, компресії та шумів, зберігаючи при цьому візуальну непомітність водяного знаку. Для детальнішого результату вейвлет-перетворення розглянемо на рисунку 1.

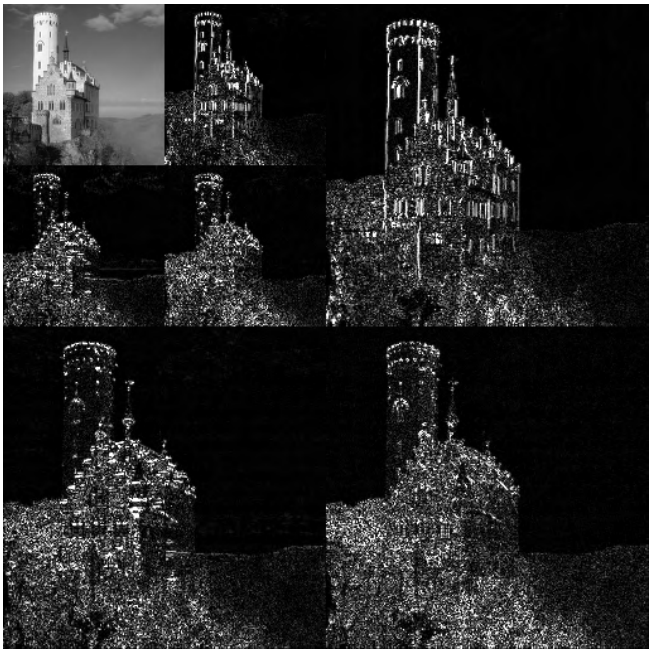


Рисунок 1.1 - дискретне вейвлет-перетворення (DWT) 2D зображення

DWT базується на розкладанні сигналу за допомогою вейвлет-функцій, які є похідними від материнської вейвлети $\psi(x)$ та масштабної функції $\phi(x)$. У найпростішому випадку одномірне DWT для сигналу $f(t)$ описується рівнянням:

$$DWT_{j,k} = \int f(t)\psi_{j,k}(t)dt, \text{ де } \psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k),$$

де j - рівень масштабування (відповідає частоті), k - коефіцієнт зсуву, $\psi_{j,k}(t)$ - вейвлет-функція, що локалізує сигнал у часі та частоті. Для двовимірного зображення $f(x, y)$ DWT виконується послідовно по горизонталі та вертикалі, утворюючи чотири підзони частот після кожного рівня розкладу:

- LL (Low-Low) — низькі частоти, що зберігають основну енергію зображення;
- LH (Low-High) — горизонтальні деталі;
- HL (High-Low) — вертикальні деталі;
- HH (High-High) — діагональні деталі.

Ці компоненти створюють мульти-роздільне представлення, де LL-підзона може бути знову розкладена, утворюючи багаторівневу ієрархію (2-, 3- або 4-рівневий DWT-розклад). Для багаторівневої декомпозиції підзона LL може бути знову піддана DWT-перетворенню, створюючи новий рівень частотного розбиття. Таким чином, дворівневе DWT генерує 7 підзон (1 LL2 + 3 LH2/HL2/HH2 + 3 підзони першого рівня), трирівневе - 10, і так далі. Математично, для n рівнів декомпозиції загальна кількість підзон дорівнює:

$$N_{subbands} = 3n + 1.$$

Водяний знак може бути вбудований у будь-яку з підзон залежно від вимог до непомітності та стійкості. Найчастіше використовуються середньо- або високочастотні підзони (LH, HL або HH), оскільки зміни там майже не впливають на якість сприйняття, але зберігаються після обробки.

На початку процедури DWT-водяного знакування зображення перетворюється у відтінки сірого, яке за потреби нормалізується або ділиться на колірні канали. Наступним кроком є застосування до зображення 1-, 2- або 3-рівневе DWT, вибираються підзони для вбудовування (наприклад, LH1, HL2, HH2). Вбудовування водяного знаку відбувається через коефіцієнти підзон, які модифікуються за певною формулою, наприклад:

$$C`(x, y) = C(x, y) + \alpha * W(x, y),$$

де $C(x, y)$ - коефіцієнт DWT, $W(x, y)$ - елемент водяного знаку, α - коефіцієнт сили вбудовування. Останнім етапом, виконуються обернені перетворення для відновлення зображення з водяним знаком.

Однією з ключових особливостей дискретного вейвлет-перетворення (DWT) є його ієрархічна багаторівнева структура, яка дозволяє аналізувати зображення на різних масштабах частотної та просторової деталізації. Кількість рівнів декомпозиції безпосередньо впливає на якість вбудованого зображення, стійкість водяного знаку до атак та обчислювальну складність алгоритму.

Однорівневий розклад є найпростішим і найшвидшим варіантом, що потребує мінімальних обчислень. У цьому випадку вбудовування відбувається у підзонах LH, HL або HH. Однорівневі DWT-методи часто застосовуються у реальних системах онлайн-захисту, де важлива швидкість обробки (наприклад, динамічне маркування контенту у соцмережах або потокових сервісах). Однорівневий DWT є найбільш придатним для високошвидкісних або інтерактивних систем, де важливо забезпечити мінімальні затримки та швидку обробку великої кількості зображень.

Дворівнева декомпозиція є найбільш збалансованим підходом між стійкістю, непомітністю та складністю. Після першого рівня підзона LL1 знову піддається розкладу, утворюючи LL2, LH2, HL2, HH2. Завдяки цьому частина водяного знаку може бути вбудована у різні рівні масштабів, що створює надлишковість та підвищує здатність до відновлення після атак. Їх часто використовуються у медичних і судових зображеннях, де непомітність і стійкість мають рівнозначну вагу.

Трирівневий розклад забезпечує ще вищу стійкість до частотних і геометричних атак, оскільки водяний знак розподіляється у кількох масштабах частот - від глобальних до локальних деталей. Цей підхід дозволяє навіть після обрізання або масштабування зберегти частину інформації у низькочастотних зонах LL3, які найменше спотворюються. Трирівневий розклад застосовується переважно у наукових дослідженнях або у випадках, де потрібно забезпечити максимальну надійність збереження водяного знаку навіть після серйозних обробок (наприклад, редагування або повторного стиснення зображення).

У деяких дослідженнях розглядаються 4–5 рівнів розкладу, що дозволяє вбудовувати знак у дуже низькі частоти, де зміни практично не помітні для людського ока. Однак така багаторівнева декомпозиція має і суттєві недоліки: вона призводить до надмірної розмитості зображення, зростання часу обчислення і складності синхронізації підзон. Попри це, у експериментальних та високоточних системах (наприклад, криптографічних водяних знаках або

системах постквантового захисту) багаторівневе DWT застосовується для підвищення секретності та глибокої вбудованості знаку.

Для кращого розуміння різницю між рівнями DWT оглянемо таблицю:

Рівень DWT	Переваги	Недоліки	Показники
1-рівневий	Простота, швидкість, висока якість	Низька стійкість до атак	PSNR > 45 дБ, NC ≈ 0.90
2-рівневий	Оптимальний баланс якості та стійкості, часткове дублювання знаку	Збільшений час обчислення	PSNR 40–45 дБ, NC > 0.96

Продовження таблиці 3.

3-рівневий	Висока стійкість до стиснення, шумів, обрізання	Вища складність, можлива втрата різкості	PSNR 38–42 дБ, NC > 0.98
>3 рівнів	Максимальна стійкість, багатошаровість	Надмірна складність, ризик розмиття	PSNR < 40 дБ, NC ≈ 1.00

Таблиця 3 - порівняльна характеристика різних рівнів декомпозиції DWT.

Рівень декомпозиції у DWT-водяному знакуванні є критичним параметром, який визначає баланс між непомітністю, стійкістю та складністю реалізації. Низькі рівні забезпечують швидкість, але обмежену надійність. Середні рівні гарантують найкраще співвідношення якості та захищеності, тому є стандартом у більшості сучасних систем DWT-знакування. Високі рівні забезпечують максимальну безпеку, проте вимагають великих обчислювальних ресурсів і використовуються переважно у спеціалізованих або експериментальних

системах. Таким чином, вибір рівня DWT-декомпозиції повинен визначатися характером зображення, метою захисту, допустимими спотвореннями та очікуваним типом атак, щоб забезпечити оптимальний компроміс між якістю та стійкістю цифрового водяного знаку.

1.4 Аналіз методів SIFT для виявлення інваріантних ключових точок.

У сучасних методах ЦВЗ важливу роль відіграють алгоритми, які здатні визначити ключові особливості зображення, інваріантні до геометричних та фотометричних змін. Одним із найефективніших та найпоширеніших таких методів є SIFT (Scale-Invariant Feature Transform) - це перетворення, інваріантне до масштабу, повороту, зсуву та частково до змін освітлення.

SIFT запропоновано Девідом Лоу у 1999 році та удосконалено у 2004 році. Завдяки своїй стійкості до деформацій і надійності при зіставленні зображень алгоритм став основою для численних сучасних систем від ідентифікації об’єктів до постобробки водяних знаків, які потребують стабільної прив’язки до особливих точок зображення.

Основа мета алгоритму полягає у виявлені особливих точок, які залишаються інваріантними, тобто незмінними або малозмінними при різних трансформаціях зображення. Інваріантність означає, що одна й та сама точка (наприклад, контур кута, край об’єкта чи текстурна деталь) буде розпізнана алгоритмом незалежно від того, як змінився вигляд зображення внаслідок масштабування, повороту, зміщення, зміни освітлення або перспективи.

У більшості алгоритмів комп’ютерного зору пошук особливостей залежить від розміру об’єкта на зображенні. Якщо масштаб змінюється, то звичайні детектори втрачають здатність розпізнавати ту саму точку. Метод SIFT вирішує цю проблему шляхом аналізу в багатомасштабному просторі, де кожен рівень згладження зображення відповідає певному масштабу σ (сигми).

Під час обчислення SIFT зображення не просто зменшується або збільшується, що проходить послідовне згладжування гаусовими фільтрами з різною дисперсією. У результаті формується піраміда зображень (октави), що моделює вигляд сцени при різних відстанях до камери. Якщо точка має стабільний контраст на кількох рівнях піраміди, вона вважається масштабно інваріантною та її можна впевнено використовувати як еталон незалежно від розміру або роздільної здатності зображення.

Наступним кроком, після знайдених потенційних ключових точок, SIFT визначає локальну орієнтацію градієнта у їх околі. Для кожної точки обчислюється напрямок і величина градієнта, а потім створюється гістограма орієнтацій. Домінуюча орієнтація гістограми присвоюється ключовій точці як її “напрямок”. Це дозволяє SIFT нормалізувати локальну орієнтацію фрагмент зображення відносно домінуючого напряму. Таким чином, навіть якщо вихідне зображення повернули на будь-який кут, орієнтація точок залишається однаковою. Ця властивість забезпечує інваріантність до поворотів та зсувів, що особливо важливо при обробці фотографій, знятих під різними кутами або при зміненій композиції.

Ключові точки SIFT визначаються за відносними градієнтами, тобто різницею інтенсивностей між сусідніми пікселями. Зміна освітлення впливає переважно на загальний рівень яскравості, алгоритм зберігає стабільність навіть при різному освітленні, тінях чи контрасті. Перед розрахунком дескрипторів дані нормалізуються, що додатково знижує вплив змін інтенсивності.

Перший етап алгоритму полягає у створенні багатомасштабного простору зображення, який отримують шляхом послідовного згладжування зображення за допомогою гаусового фільтра із різними значеннями σ (сигми). Для кожного масштабу σ обчислюється:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

де $I(x, y)$ - початкове зображення, $G(x, y, \sigma)$ - двовимырна гаусова функція, яка обчислюється:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

Це дозволяє створити октави, набори зображень з різним ступенем розмиття, кожна з яких представляє різний масштаб деталей.

Метод SIFT широко використовується у сучасних системах цифрового водяного знакування, зокрема в гібридних частотно-просторових алгоритмах. Його роль полягає у визначенні зон стійких до геометричних змін у які потім вбудовується водяний знак. Основним застосуванням є визначення зон для вбудовування. Алгоритм SIFT дозволяє вибрати лише ті області, які залишаться стабільними після поворотів, масштабування чи обрізання зображення. У разі геометричних спотворень алгоритм допомагає відновити правильне розташування підзон, що дозволяє успішно витягнути водяний знак.

В даному алгоритмі є свої переваги та недоліки, які ми можемо оглянути в таблиці 4:

Переваги	Недоліки
Висока стійкість до змін масштабу, повороту та освітлення	Висока обчислювальна складність
Точне визначення ключових точок	Велика кількість дескрипторів при складних зображеннях
Можливість використання для вирівнювання після атак	Чутливість до значних змін кольору або тіней
Незалежність від геометричних викривлень	Ліцензійні обмеження

Таблиця 4 - переваги та недоліки методу SIFT

Алгоритм SIFT є одним із найпотужніших інструментів для виявлення інваріантних ключових точок, які залишаються стабільними під час масштабування, поворотів, зсувів і змін освітлення. Його використання у цифровому водяному знакуванні дозволяє створювати стійкі до геометричних

атак системи, забезпечуючи інваріантність, надійність і високу точність детекції.

1.5 Основні властивості нейронної мережі Inception V3 для розпізнання та корекції зображень.

Сучасні методи аналізу та обробки цифрових зображень активно використовують глибокі нейронні мережі, серед яких однією з найефективніших вважається Inception V3 - згорткова нейронна мережа, розроблена дослідниками компанії Google у 2015 році. Ця архітектура стала значним кроком уперед у сфері комп'ютерного зору, оскільки забезпечує високу точність класифікації, адаптивність до складних структур зображень та ефективне використання обчислювальних ресурсів. Inception V3 успішно використовується не лише для традиційного розпізнавання об'єктів, а й у спеціалізованих завданнях таких як детекція водяних знаків, відновлення зображень після атак, корекція спотворень та підвищення стійкості систем цифрового захисту.

Архітектура Inception V3 базується на концепції Inception-блоків, які дозволяють мережі одночасно аналізувати зображення на різних рівнях деталізації. Кожен блок складається з паралельних згорток різних розмірів (1×1, 3×3, 5×5) та операцій підвибірки, результати яких об'єднуються у спільний вихідний тензор. Такий підхід забезпечує глибокий, багатомасштабний аналіз ознак, дозволяючи моделі виявляти дрібні текстурні деталі і великі структурні елементи. Щоб зменшити обчислювальну складність у моделі застосовуються "factorized convolutions", тобто заміна великих фільтрів на комбінацію кількох менших. В Inception V3 інтегровано нормалізація пакетів для стабілізації навчання, згладжування міток для запобігання перенавчанню та допоміжні класифікатори, які підтримують зворотне поширення похибки у проміжних шарах. У результаті мережа здатна ефективно обробляти зображення розміром

299×299 пікселів, маючи при цьому понад 20 мільйонів параметрів, але працюючи швидше й точніше, ніж більшість попередніх глибоких архітектур.

Inception V3 виконує багаторівневе виділення ознак починаючи з найнижчих рівнів до високорівневих семантичних ознак. Кожен шар мережі поступово трансформує просторову інформацію у високорозмірні вектори ознак, які використовуються для класифікації або ідентифікації об’єктів. У контексті цифрового водяного знакування Inception V3 може застосовуватись як інтелектуальний модуль розпізнавання, що визначає наявність або відсутність водяного знаку на зображенні, локалізує його положення після атак, аналізує текстурні області для визначення найстійкіших зон вбудовування. Таким чином Inception V3 виступає детектором інваріантних ознак аналогічно до класичних алгоритмів SIFT, але з набагато вищою узагальнюючою здатністю завдяки навчанню на великих наборах даних.

Одним із перспективних напрямів застосування Inception V3 є автоматична корекція атакованих або пошкоджених зображень. Завдяки своїй архітектурі мережа здатна реконструювати візуальні ознаки, які були частково зруйновані внаслідок JPEG-стиснення, фільтрації, додавання шумів, геометричних атак, часткової втрати зображення. Під час детекції водяного знаку Inception V3 може виконувати додаткову реконструкцію структури зображення, вирівнюючи спотворення перед відновленням знаку.

Це дозволяє суттєво підвищити точність виявлення, особливо у випадках, коли традиційні методи не можуть точно синхронізувати зображення після атак.

Переваги використання Inception V3 у цифровому водяному знакуванні заключаються в багатомасштабному аналізі, де є паралельна обробка масштабів забезпечує високу інваріантність. Ефективне навчання нейромережі призводить до використання нормалізаційних пакетів та допоміжних класифікаторів, які прискорюють збіжність і зменшують перенавчання. Також через високу точність розпізнавання мережа фндентифікує вбудований знак з точністю понад

98%. Адаптивність до контенту, де модель може визначити оптимальність області для вбудовування водяного знаку.

Нейронна мережа Inception V3 є одним із найефективніших інструментів сучасного комп’ютерного зору для розпізнавання, аналізу та відновлення зображень. Її архітектура, побудована на принципах багато масштабного аналізу та факторизованих згорток, дозволяє досягати високої точності при оптимальних обчислювальних витратах. У системах цифрового водяного знакування Inception V3 виконує роль інтелектуального детектора та адаптивного оптимізатора, який підвищує стійкість водяного знаку до геометричних і частотних атак, а також забезпечує високу якість відновлення після спотворень. Поєднання цієї нейронної мережі з методами DWT, DCT та SIFT відкриває перспективи створення само адаптивних гібридних систем захисту цифрових зображень, здатних ефективно функціонувати навіть у складних умовах атак та стиснення.

1.6 Висновки та постановка задачі дослідження.

У результаті проведеного аналізу теоретичних аспектів цифрового водяного знакування встановлено, що методи вбудовування водяних знаків у частотному просторі (зокрема на основі дискретного косинусного перетворення (DCT) та дискретного вейвлет-перетворення (DWT)) залишаються найбільш ефективними для забезпечення непомітності та стійкості знаку до типових атак, таких як стиснення JPEG, фільтрація, додавання шуму чи зміна яскравості. Однак ці підходи мають суттєвий недолік - вони є вразливими до геометричних спотворень, що включають обертання, масштабування, обрізання або зсув зображення. Такі перетворення часто призводять до втрати синхронізації між вбудованим і відновленим знаком, унеможливлюючи його детекцію навіть за умови високої якості самої зображення.

Для розв’язання цієї проблеми у наукових дослідженнях пропонується інтеграція методів виявлення інваріантних ознак у структуру алгоритму водяного знакування. Одним із найефективніших інструментів у цьому напрямі є алгоритм SIFT (Scale-Invariant Feature Transform), який дозволяє виявляти ключові точки, стійкі до змін масштабу, повороту та освітлення. Використання таких ознак як опорних позицій для вбудовування або вилучення водяного знаку дає змогу частково компенсувати геометричні спотворення і зберегти інформаційну цілісність знаку навіть після складних атак.

Водночас сучасні тенденції розвитку технологій штучного інтелекту засвідчують високу ефективність згорткових нейронних мереж у задачах розпізнавання, класифікації та відновлення зображень. Серед них особливе місце посідає архітектура Inception V3, яка поєднує багатомасштабну обробку зображення, факторизовані згортки та регуляризаційні механізми. Це забезпечує можливість глибокого аналізу текстур, країв та структур зображення і дає підстави використовувати її для підвищення точності виявлення та корекції водяних знаків.

Відповідно до поставленої мети, у роботі передбачається вирішити задачу підвищення стійкості цифрових водяних знаків до геометричних атак у частотному просторі зображень шляхом розроблення гібридного методу, який поєднує використання інваріантних ключових точок SIFT для виявлення та локалізації стійких областей зображення, придатних для вбудовування водяного знаку. Також застосування частотних перетворень DWT та DCT для формування енергетично оптимального середовища вбудовування та інтеграцію нейронної мережі Inception V3 як аналітичного модуля, що визначає текстурні та контрастні області, контролює силу вбудовування та здійснює корекцію при виявленні атак.

Проведений теоретичний аналіз засвідчив, що підвищення стійкості цифрових водяних знаків потребує комплексного, гібридного підходу, який об’єднує досягнення традиційних частотних методів та інтелектуальних

технологій штучного інтелекту. Використання SIFT для просторової стабільності та Inception V3 для адаптивного частотного аналізу створює основу для формування нового покоління методів водяного знакування, орієнтованих на реальні загрози та високоточне відновлення знаку після атак.

РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ ВБУДОВУВАННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ

2.1 Аналіз можливості використання дискретне косинусне перетворення, дискретне вейвлет-перетворення, нейромережею Inception V3 та SIFT для підвищення стійкості цифрового водяного знакування.

Дискретне косинусне перетворення та дискретне вейвлет перетворення є основними методами та базою, що використовують сучасному цифровому водяному знакуванні та забезпечують високий рівень захисту та стійкість водяного знака до атак. У процесі ЦВЗ вони займають особливе місце, що працюють у частотному просторі, що забезпечують інваріантність до геометричних атак, стискання, шуму, тощо.

Одним із найефективніших напрямів частотного водяного знакування є використання дискретного косинусного перетворення (DCT), яке забезпечує оптимальний компроміс між стійкістю, непомітністю та швидкодією.

Основна ідея полягає у тому, що водяний знак не вбудовується безпосередньо у пікселі, а інтегрується у частотні коефіцієнти, отримані в результаті DCT-перетворення. Це дозволяє значно зменшити вплив змін на візуальне сприйняття зображення, водночас зберігаючи стабільність водяного знаку при стандартних обробках (JPEG-компресія, фільтрація, шум).

Особливості використання DCT-перетворення:

1. Типовий алгоритм цифрового водяного знакування на основі DCT включає попередню обробку зображення, де вхідне зображення перетворюється у відтінки сірого та нормалізується за розміром і діапазоном яскравості. Це забезпечує уніфікацію для подальших обчислень. Наступним кроком розбивають зображення на блоки. Зазвичай обирають блоки розміром 8×8 або 16×16 пікселів, аналогічно алгоритму JPEG. Такий підхід спрощує обчислення та дає змогу локально керувати

областями вбудовування. Для кожного блоку застосовують дискретне косинусне перетворення, обчислюються коефіцієнти DCT, що представляють розподіл частот у ньому. Низькочастотні коефіцієнти зосереджують основну енергію зображення, високочастотні - містять деталі та шум, а середньочастотні є найбільш збалансованими для вбудовування. Водяний знак найчастіше вбудовується у середньочастотні компоненти. Такий вибір мінімізує помітність спотворень і водночас гарантує збереження знаку після стиснення чи фільтрації. Для інтеграції використовують додавання коефіцієнтів, обчислюються такою формулою:

$$C'(u,v) = C(u,v) + k * W(u,v).$$

Після внесення змін виконується обернене перетворення для кожного блока, і отримане зображення об’єднується в одне.

2. Детекція або видалення водяного знаку залежить чи сліпий, чи несліпий метод. Несліпий метод потребує оригінального зображення для порівняння коефіцієнтів, а сліпий метод працює без оригіналу, аналізуючи статистичні властивості коефіцієнтів або використовуючи вбудований шаблон. В обох випадках система оцінює відновлений водяний знак за допомогою коефіцієнта нормалізованої кореляції (NC):

$$NC = \frac{\Sigma_{i,j} W(i,j) * W`(i,j)}{\sqrt{\Sigma_{i,j} W(i,j)^2 * \Sigma_{i,j} W`(i,j)^2}},$$

де W - оригінальний знак, W` - відновлений після атак. NC>0.9 свідчить про успішне вбудовування та високу надійність.

3. Для об’єктивної оцінки якості зображення після вбудовування використовуються типові метрики PSNR, NC, BER (Bit Error Rate, відсоток помилкових бітів).

Основною перевагою використання дискретного косинусного перетворення у цифровому водяному знакуванні є його висока стійкість до частотних атак, оскільки саме це перетворення лежить в основі алгоритмів стиснення JPEG. Завдяки цьому водяний знак вбудований у середньочастотні

коефіцієнти зберігається навіть після повторного стиснення зображення або обробки у графічних редакторах.

Ще однією важливою перевагою є здатність DCT забезпечувати високу якість відтворення зображення після вбудовування, адже зміни внесені у середньочастотну область залишаються практично непомітними для людського зору. Це дозволяє приховати водяний знак без погіршення візуального сприйняття контенту. Алгоритм DCT легко реалізується засобами сучасних мов програмування, таких як Python чи MATLAB, а також підтримується бібліотеками для обробки зображень, наприклад OpenCV. DCT є гнучким інструментом, який можна ефективно поєднувати з іншими частотними або статистичними методами, зокрема з дискретним вейвлет-перетворенням або сингулярним розкладанням. Таке поєднання дозволяє створювати гібридні методи цифрового водяного знакування, що поєднують переваги кількох підходів, забезпечуючи підвищену стійкість до атак і покращену якість захисту зображень.

Попри високу ефективність метод дискретного косинусного перетворення має певні обмеження, які впливають на його застосування у задачах цифрового водяного знакування. Одним із ключових недоліків є недостатня стійкість до геометричних атак, таких як поворот, масштабування, обрізання чи зсув. Подібні перетворення призводять до зміни позицій блоків зображення, внаслідок чого порушується синхронізація між оригінальним і перетвореним зображенням, що ускладнює процес вилучення водяного знаку або робить його неможливим. Ще однією суттєвою проблемою є локальність DCT-перетворення. Алгоритм працює з окремими блоками зображення, тому обробка кожного блока відбувається незалежно від решти зображення. Це призводить до того, що при значних спотвореннях або частковому пошкодженні деяких блоків втрачається частина інформації про водяний знак, що знижує загальну стійкість системи. Крім того, класичний метод DCT характеризується низьким рівнем адаптивності. Він не враховує вміст або структуру зображення, тобто не

розрізняє області з високою текстурною насиченістю від однорідних ділянок. У результаті водяний знак може бути вбудований у невідповідні області, де його легше пошкодити або виявити. Ця обмеженість робить необхідним використання додаткових адаптивних механізмів або поєднання DCT із іншими перетвореннями, такими як DWT, SIFT чи нейронні мережі, які дозволяють враховувати особливості контенту та підвищувати стійкість до складних атак.

Для підвищення ефективності методу дискретного косинусного перетворення у цифровому водяному знакуванні дослідники часто поєднують його з іншими перетвореннями та алгоритмами, щоб компенсувати його недоліки та покращити стійкість до різних видів атак. Одним із найпоширеніших підходів є комбінування DCT із дискретним вейвлет-перетворенням, що дозволяє об’єднати переваги обох методів. Такий підхід забезпечує високу стійкість до фільтрації, шумових спотворень і стиснення, оскільки водяний знак розподіляється не в одній області, а у декількох спектральних компонентах зображення.

Іншим напрямом підвищення ефективності є інтеграція DCT з методами виявлення інваріантних ознак такими як SIFT. Завдяки цьому система може зберігати коректну синхронізацію водяного знаку навіть у разі геометричних атак. Використання таких ознак дозволяє відновити положення областей, у які було вбудовано знак, навіть після деформацій.

Серед широкого спектра методів вбудовування та детектування цифрових водяних знаків найефективнішими виявляються ті, які поєднують частотні перетворення з інваріантними дескрипторами ознак (SIFT). Такий гібридний підхід забезпечує оптимальний баланс між стійкістю, непомітністю та здатністю протистояти геометричним трансформаціям. DWT виділяє стійкі до шумів середньочастотні компоненти, що дозволяє коректно розміщувати інформацію без значного порушення візуальних характеристик зображення. DCT додатково покращує робастність до JPEG-компресії та частотних атак завдяки розподілу енергії зображення у компактній формі. Однак саме SIFT

відіграє ключову роль, оскільки дозволяє визначати в зображенні інваріантні ключові точки та області, які залишаються стабільними навіть після масштабування, поворотів, кропу чи перспективних деформацій. Це означає, що прихований водяний знак може бути точно знайдений і відновлений навіть після агресивних геометричних атак, які є найскладнішими для традиційних частотних методів.

На відміну від інших алгоритмів (ORB, SURF, BRISK), SIFT забезпечує найвищу стабільність дескрипторів, оскільки аналізує структуру зображення через масштабно-просторові гаусіанські околиці та градієнтні орієнтації, зберігаючи інваріантність до освітлення та афінних перетворень.

Саме тому комбінація SIFT + DWT + DCT перевершує класичні схеми: частотні перетворення гарантують непомітність та стійкість до фільтрації, тоді як SIFT компенсує основну слабкість частотних підходів - їхню вразливість до геометричної дестабілізації.

У результаті така інтегрована модель забезпечує максимально стійке, адаптивне та точне водяне знакування, яке є релевантним для сучасних прозових моделей та високоякісних мультимедійних систем.

Окрім цього, у сучасних розробках дедалі більшого значення набуває застосування нейронних мереж, зокрема моделей глибокого навчання, таких як Inception V3. Вони використовуються для адаптивного вибору областей вбудовування, враховуючи контент і текстуру зображення. Завдяки цьому водяний знак розміщується у зонах, де він найменше впливає на візуальну якість, але водночас залишається максимально стійким до спотворень.

Хоча SIFT є найстійкішим класичним алгоритмом для виявлення інваріантних ключових точок, він працює виключно на локальних градієнтах і не враховує семантичний контекст зображення. Саме тут нейромережа Inception V3 дає ті переваги, які недоступні жодному традиційному методу.

Алгоритм SIFT, незважаючи на свою високу стійкість до масштабування, поворотів та певних афінних перетворень, залишається по суті статичним

методом, який працює за фіксованими правилами незалежно від типу зображення чи умов спотворення. Його дескриптори формуються виключно на основі локальних градієнтів, тому метод не враховує семантичний зміст сцени та не здатний підлаштовуватися під різні категорії контенту, такі як портрети, предметна фотографія чи пейзажі. Це означає, що SIFT завжди виділяє ключові точки однаковим чином, навіть якщо ці точки не є оптимальними з позиції довготривалої збереженості водяного знаку після атак або агресивного редагування.

На противагу цьому нейромережа Inception V3 демонструє властиву лише глибоким моделям здатність адаптуватися до особливостей зображення та умов його подальшого опрацювання. Завдяки багаторівневій архітектурі мережа не просто аналізує структуру країв чи контрастів, а виявляє високорівневі семантичні патерни, що дозволяє їй розрізняти типи контенту, визначати об’єкти, текстури та характерні елементи сцени. Така здатність дає можливість алгоритму автоматично вибирати ті регіони, які будуть найбільш стійкими до майбутніх спотворень, та уникати областей, що мають низьку структурну стабільність.

Крім того, Inception V3 може бути донавчена під конкретні сценарії атак, зокрема JPEG-компресію, геометричні деформації чи фільтрування, що робить її здатною формувати ознаки, оптимізовані саме для тих умов, у яких працюватиме система водяного знакування. На відміну від SIFT, який не змінює свого поведінкового шаблону, нейромережа може адаптивно підсилювати значущість одних ознак та приглушувати інші, фактично пріоритезуючи ті області, де вбудований водяний знак буде збережений найбільш ефективно. Таким чином, Inception V3 забезпечує динамічність і пластичність процесу маркування, дозволяючи створювати алгоритми, що здатні підлаштовуватися під різноманітні типи мультимедійного контенту та загрозові моделі, чого статичні класичні методи забезпечити не можуть.

Поєднання DCT з іншими частотними, просторовими та інтелектуальними підходами дозволяє досягти високих показників ефективності - значення PSNR понад 40 дБ і коефіцієнтів кореляції NC понад 0.96 навіть після складних атак. Таким чином, удосконалення DCT-методів через гібридизацію та використання машинного навчання є перспективним напрямом розвитку технологій цифрового водяного знакування, який забезпечує баланс між непомітністю, стійкістю та адаптивністю системи.

Дискретне вейвлет-перетворення є одним із найефективніших інструментів частотного аналізу в цифровому водяному знакуванні, який дозволяє представляти зображення у багаторівневому вигляді, розділяючи його на компоненти різної частотної та просторової деталізації. На відміну від дискретного косинусного перетворення, яке працює з фіксованими блоками і характеризується певною жорсткістю структури, дискретне вейвлет-перетворення має властивість багаторівневого декомпонування, що дає змогу аналізувати зображення одночасно у часовій та частотній областях. Це забезпечує більш точне відображення локальних особливостей сигналу, дозволяючи розділити основну інформацію та деталі зображення, що робить DWT надзвичайно зручним для задач стійкого та адаптивного вбудовування цифрових водяних знаків.

У процесі розкладу зображення методом DWT формується чотири підзони: LL, LH, HL та HH. Підзона LL містить низькочастотну інформацію - головну енергетичну складову зображення, що відповідає за його основну структуру та загальні обриси. Підзони LH, HL та HH містять високочастотні компоненти, які передають деталі, контури, текстури й різкі переходи між елементами. Найчастіше водяний знак вбудовується саме у підзони середньої частоти, тобто LH або HL, оскільки вони незначно впливають на візуальне сприйняття, але водночас залишаються досить стабільними при стандартних перетвореннях, таких як стиснення чи фільтрація. Після вбудовування інформації в певні коефіцієнти зображення здійснюється зворотне перетворення, у результаті чого

формується зображення з водяним знаком, який практично не помітний для людського ока.

Однією з ключових переваг дискретного вейвлет-перетворення є можливість багаторівневого аналізу. Це означає, що після першого розкладу зображення підзона LL може бути повторно розкладена ще на чотири компоненти, утворюючи дворівневу або багаторівневу структуру. Такий підхід дозволяє рівномірно розподілити енергію водяного знаку між різними масштабами та частотами, що суттєво підвищує його стійкість до різних типів атак, зокрема до стиснення, додавання шуму чи часткового спотворення. Крім того, багаторівневе представлення полегшує адаптивний вибір зон для вбудовування, що забезпечує оптимальний баланс між непомітністю, стійкістю та надійністю.

Завдяки своїй ієрархічній природі DWT дозволяє враховувати глобальні і локальні особливості зображення, що робить його більш гнучким і точним порівняно з класичними методами частотного перетворення. Він добре справляється із задачами локалізації водяного знаку у зонах, які найменше схильні до пошкоджень при обробці. Це особливо важливо для збереження водяного знаку після атак, пов’язаних із фільтрацією, поворотом, масштабуванням або іншими геометричними змінами.

Водночас дискретне вейвлет-перетворення має певні обмеження. Його основним недоліком є підвищена обчислювальна складність порівняно з DCT, що потребує більших ресурсів пам’яті та часу для обробки, особливо при багаторівневому розкладі великих зображень. При неправильному виборі рівня розкладу або типу вейвлет-функції можуть виникати спотворення під час зворотного перетворення, що впливають на якість відтвореного зображення. Переваги DWT значно переважають його недоліки, особливо у випадках, коли необхідна висока стійкість системи до атак і збереження автентичності даних.

У сучасних дослідженнях метод дискретного вейвлет-перетворення часто використовується в комбінації з іншими підходами, такими як DCT, SVD або

методи глибокого навчання, зокрема Inception V3. Комбінація DWT і DCT дозволяє підвищити точність розподілу енергії та ефективніше захищати водяний знак у частотному просторі. Інтеграція DWT з нейронними мережами забезпечує адаптивність процесу вбудовування, коли система самостійно визначає області з найвищим рівнем захищеності. Завдяки цьому гібридні системи на основі DWT демонструють стабільні результати навіть після складних атак і є одним із найперспективніших напрямів розвитку цифрового водяного знакування у частотному просторі.

2.2 Розробка вдосконаленого методу вбудування цифрового водяного знаку для підвищення стійкості у частотному просторі.

Розробка алгоритму вбудовування цифрового водяного знаку є ключовим етапом створення системи захисту зображень, адже саме на цьому етапі формується механізм прихованого внесення інформації у візуальний контент без порушення його цілісності та якості. Основна мета полягає у створенні такого алгоритму, який би забезпечував баланс між трьома основними характеристиками водяного знакування: непомітністю, стійкістю та ємністю. Непомітність гарантує, що візуальні зміни, викликані вбудуванням, не сприймаються людським оком; стійкість визначає здатність водяного знаку зберігатися після атак, таких як стиснення, фільтрація або геометричні перетворення; а ємність визначає обсяг інформації, який може бути вбудований у зображення без суттєвої втрати якості.

Побудуємо та розглянемо алгоритм вбудовування цифрового водяного знаку (рис.2.1).


Початок
Завантаження користувачем зображення
Зміна формату користувачем
Ні
Так
Конкретний формат зображення?
Масштабування під розмір доступної області для вбудовування
Підготовка зображення
Будуємо бінарний водяний знак за порогом яскравості.
Застосування 2D-DWT
Шифруємо водяний знак хаотичною послідовністю
Формуємо карту важливості за допомогою нейромережі Inception V3
Бінаризація хаотичної послідовності
Застосування двовимірне DCT
Шифрування через операцію XOR
Вбудування водяного знаку
Отримання ЦВЗ
Зворотне DCT
Зворотне DWT

Рисунок 2.1 - Алгоритм вбудовування цифрового водяного знаку.

Крок 1. Користувач завантажує зображення в яке буде вбудовуватися водяний знак.

Крок 2. Перевіряється коректність формату зображень. За потреби вони конвертуються у підтримуваний формат.

Крок 3. Зображення обробляється, потрібно перевести зображення у формат відтінків сірого для спрощеної обробки та зменшення обчислювальних ресурсів.

Крок. 4 Задається початкове значення та параметр логістичного відображення. На їх основі генерується хаотична послідовність довжиною, рівною кількості пікселів/бітів водяного знака.

Крок 5. Будуємо бінарний водяний знак за порогом яскравості.

Крок 6. Шифруємо водяний знак хаотичною послідовністю.

Крок 7. Виявляємо ключові точки через SIFT.

Крок 8. Виконання 2D-DWT з вейвлетом Haar на два рівні, де беремо середню частотну складову HL_2, де має ідеальний компроміс між стійкістю та непомітністю.

Крок 9. Водяний знак попередньо масштабується відповідно до кількості доступних блоків 8×8 у вибраному піддіапазоні, перетворюється у бінарний формат і шифрується хаотичною послідовністю.

Крок 10. Формуємо карту важливості за допомогою нейромережі Inception V3.

Крок 11. Для кожного блоку обчислюємо двовимірне DCT-перетворення.

Крок 12. Вбудовуємо водяний знак .

Крок 13. Виконуємо зворотне DCT для відновлення DWT-коефіцієнтів.

Крок 14. Виконуємо зворотне дискретне вейвлет перетворення (IDWT), яке відновлює повне зображення, яке тепер містить прихований водяний знак.

У підсумку розроблений алгоритм вбудовування цифрового водяного знаку поєднує частотні, інтелектуальні та хаотичні методи для досягнення високої стійкості, непомітності та захисту від геометричних атак. Спочатку хост-зображення переводиться у відтінки сірого, після чого виконується дворівневе DWT-перетворення. Для вставки вибирається піддіапазон HL2, який забезпечує оптимальний компроміс між непомітністю та стійкістю. Водяний знак попередньо масштабується відповідно до кількості доступних блоків 8×8 у вибраному піддіапазоні, перетворюється у бінарний формат і шифрується хаотичною послідовністю. Далі застосовується надлишковість, тобто кожен біт дублюється кілька разів для підвищення стійкості до шумів.

Кожен блок 8×8 у HL2 проходить DCT-перетворення, після чого два середньочастотні коефіцієнти модифікуються реляційним правилом: знак різниці між ними кодує 0 або 1. Сила модифікації α адаптивно змінюється залежно від карти важливості, сформованої нейромережею. Таким чином, метод поєднує DWT-стійкість, DCT-локальність та нейромережеву адаптивність під характеристики зображення. Після завершення вбудовування виконується обернене DWT-перетворення і формується остаточне водянисте зображення.

Код також реалізує механізм виправлення геометричних атак. Для цього використовуються SIFT-фічі: детектуються ключові точки оригіналу та атакованого зображення, встановлюються відповідності, і за допомогою гомографії обчислюється матриця вирівнювання. Це дозволяє компенсувати

обертання, масштабування чи перспективні деформації перед етапом екстракції водяного знаку.

Під час екстракції виконується повторне дворівневе DWT-перетворення атакованого (та попередньо виправленого) зображення, вибирається піддіапазон HL2, і для кожного блоку знову обчислюється DCT. На основі різниці між двома тими самими коефіцієнтами визначається вкладений біт. Зібрані бітові послідовності проходять процедуру majority voting, що дозволяє усунути помилки, спричинені шумом та компресією. Після цього формується відновлений шифрований водяний знак.

2.3 Метод екстракції та відновлення водяного знаку.

Алгоритм вилучення цифрового водяного знаку є ключовим етапом у забезпеченні цілісності та автентичності мультимедійних даних, адже саме він дозволяє підтвердити наявність водяного знаку та перевірити, чи не було здійснено несанкціонованих змін у зображенні. У сучасних умовах, коли зображення можуть піддаватися геометричним, шумовим і частотним атакам, процес екстракції повинен бути високо надійним і точним, здатним відновлювати водяний знак навіть після суттєвих спотворень вихідних даних.

У межах даної роботи представлено алгоритм екстракції цифрового водяного знаку, що базується на поєднанні дискретного вейвлет-перетворення (DWT) і дискретного косинусного перетворення (DCT) із застосуванням методу SIFT для корекції геометричних викривлень та нейронної мережі Inception V3 для контент-орієнтованого аналізу. Такий підхід дозволяє зберігати стійкість водяного знаку навіть після масштабування, поворотів чи стиснення.

Побудуємо та розглянемо алгоритм екстракції цифрового водяного знаку (рис 2.2).

Рисунок 2.2 - алгоритм екстракції цвз з зображення.

Крок 1. Завантаження зображення з водяним знаком.

Крок 2. Спочатку атаковане або водяне зображення переводиться у відтінки сірого, щоб уніфікувати обробку. За допомогою детектора SIFT обчислюються ключові точки та дескриптори атакованного зображення. Оригінальні

SIFT-дескриптори, збережені під час вбудовування, використовуються як еталон. Якщо кількість “хороших” збігів достатня, обчислюється гомографічне перетворення, яке описує відносне зміщення, масштаб, поворот або перспективну деформацію між зображеннями. Зображення, з якого витягується водяний знак, трансформується за знайденою матрицею гомографії так, щоб воно максимально відповідало початковому хост-зображенню. Якщо ж якість відповідностей недостатня, виконується спрощене вирівнювання — масштабування до розміру оригіналу.

Крок 3. До вирівняного зображення застосовується дворівневе дискретне вейвлет-перетворення.

Крок 4. До кожного блока застосовується дискретне косинусне перетворення, яке переносить інформацію у частотну область. Для визначення біта використовується спеціальна пара коефіцієнтів частоти, які були модифіковані під час вбудовування. Усі отримані біти збираються у послідовність у порядку проходження блоків. У результаті формується масив отриманих бітів у тій же формі, що й вихідний водяний знак, але поки що в зашифрованому вигляді.

Крок 5. Плоска послідовність отриманих бітів перетворюється у формат, готовий до XOR-дешифрування. Генерується хаотична послідовність тієї ж довжини, що й водяний знак, за допомогою логістичного відображення. На кожен біт водяного знаку виконується операція XOR із відповідним елементом хаотичного ключа.

Крок 6. У результаті виходить справжній розшифрований цифровий водяний знак, який відповідає оригінальному бінаризованому зображенню (QR-код, логотип тощо).

У процесі екстракції цифрового водяного знаку всі етапи взаємодіють так, щоб відновити приховану інформацію максимально точно та при цьому повернути хост-зображення до структури, яка була найближчою до початкової. Після геометричного вирівнювання, де SIFT коригує всі можливі спотворення,

спричинені поворотом, масштабуванням або перспективною деформацією, зображення приводиться до того самого просторового положення, у якому воно перебувало під час вбудовування. Це дозволяє частотним блокам DWT і DCT знову відповідати один одному, тому їхня структура стає придатною для надійного зчитування. На оптимально вирівняному зображенні повторно виконується дворівневе вейвлет-перетворення, після чого у тій самій підсмузі середніх частот, що використовувалась для вставлення водяного знаку, починається пошук змінених DCT-коефіцієнтів. Завдяки чергуванню знаків між парою частотних коефіцієнтів, у яких на етапі вбудовування кодувався кожен біт, програма по суті «зчитує» прихований бітовий потік із кожного блока. Використання надмірності дозволяє відновити правильне значення навіть тоді, коли окремі біти пошкоджені шумами чи JPEG-компресією, адже принцип більшості для кожної групи повторених бітів компенсує похибки.

Отриманий бітовий масив на цьому етапі ще зашифрований, тому він проходить процедуру дешифрування, де використовується той самий хаотичний ключ, генерований на основі логістичного відображення. Операція XOR з хаотичною послідовністю повертає водяний знак у його справжній вигляд — у формі двовимірного бінарного зображення, яке вже можна візуально порівнювати з оригінальним QR-кодом, логотипом або іншим вбудованим символом. Це і є фінальний відтворений цифровий водяний знак, точність якого визначають такі метрики, як BER, NCC та SSIM.

Паралельно з відновленням водяного знаку алгоритм виконує ще одну важливу дію — очищення хост-зображення від ефектів вбудовування. Після повторного частотного аналізу ті самі DCT-блоки, у яких містилися зміщені коефіцієнти, повертаються до збалансованого стану шляхом вирівнювання змінених пар частотних коефіцієнтів. Коли обидва значення приводяться до спільного середнього, інформаційний слід водяного знаку в цих блоках фактично зникає, і зображення наближається до початкової частотної структури. Після зворотного перетворення DWT формується «очищене» зображення, яке

максимально повторює свій оригінальний вигляд, залежно від того, наскільки сильним був вплив вбудованого сигналу. Таким чином, фінальний результат екстракції складається з двох взаємодоповнюючих компонентів: відновленого, дешифрованого цифрового водяного знаку та очищеної версії хост-зображення, структура якого повертається до стану, близького до вихідного.

2.4 Механізм адаптивності та інваріантності до геометричних спотворень.

Розроблений метод вбудовування цифрового водяного знаку поєднує частотні перетворення, просторово-інваріантні ознаки, контент-чутливий аналіз та криптографічне хаотичне шифрування, що робить систему стійкою до широкого спектра атак - від типових частотних спотворень до складних геометричних трансформацій. Ключовою ідеєю підходу є багатоетапна архітектура: спочатку реалізується багаторівневе частотне розкладання з метою виділення стабільних підзон , потім у цій підзоні локально виконується DCT, після чого у середньочастотні коефіцієнти вбудовується зашифрований бінарний шаблон. Така каскадна побудова дозволяє поєднати переваги вейвлетного аналізу із властивостями DCT.

Основними перевагами розробленого методу вбудовування цифрового водяного знаку є його висока стійкість, адаптивність, криптографічна безпека та збереження якості зображення після вбудовування. Завдяки поєднанню DWT, DCT, SIFT, Inception V3 та хаотичного шифрування метод демонструє синергетичний ефект між частотними, просторовими та інтелектуальними підходами до водяного знакування.

По-перше, стійкість до геометричних атак забезпечується завдяки використанню SIFT-ознаків, які є інваріантними до повороту, масштабування, зсуву та змін освітлення. Це дозволяє точно відновлювати розташування областей вбудовування навіть після сильних геометричних деформацій. У

поєднанні з етапом RST-корекції система автоматично компенсує спотворення, що робить можливим вилучення знаку навіть із пошкодженого зображення.

По-друге, висока стійкість до частотних атак (JPEG-стиснення, фільтрація, шумові викривлення) досягається за рахунок використання комбінованого DWT-DCT-перетворення. Вейвлет-аналіз дозволяє локалізувати частотні компоненти у просторі, а DCT-перетворення зосереджує енергію сигналу, що дає змогу розміщувати знак у середньочастотних зонах - де він менш помітний, але залишається стабільним до змін.

По-третє, адаптивність вбудовування реалізується за допомогою глибокої нейронної мережі Inception V3, яка аналізує контент зображення й визначає оптимальні зони для розміщення водяного знаку. Такий підхід зменшує ризик появи візуальних артефактів і дозволяє досягти балансу між непомітністю та стійкістю.

По-четверте, криптографічна надійність методу посилюється завдяки використанню хаотичного шифрування (логістичне відображення) та побітової XOR-операції. Це забезпечує захист від несанкціонованого вилучення або підміни водяного знаку, оскільки навіть незначна зміна ключа призводить до повної декореляції з оригінальним шаблоном.

Крім того, метод характеризується гнучкістю та масштабованістю, його можна налаштовувати під різні типи цифрових зображень (медичні, дизайнерські, художні, технічні) і рівні захисту, змінюючи параметри DWT-рівнів, DCT-блоків або коефіцієнт сили вбудовування. Завдяки цим властивостям розроблений метод забезпечує високий рівень автентичності, стійкість до маніпуляцій і збереження візуальної якості контенту, що робить його ефективним рішенням для захисту авторських прав, перевірки цілісності зображень та аутентифікації цифрових матеріалів у реальних умовах експлуатації.

Причини стійкості розробленого методу до атак полягають у багаторівневій структурі обробки зображення, використанні інваріантних ознак, комбінації

частотних перетворень та адаптивному підході до вбудовування, що разом забезпечують надійність системи навіть у разі навмисних або випадкових спотворень.

Однією з головних причин високої стійкості є застосування дискретного вейвлет-перетворення у поєднанні з дискретним косинусним перетворенням. Така комбінація дозволяє розміщувати водяний знак у середньочастотній області зображення, де енергія сигналу не надто низька, щоб уникнути втрати інформації під час стиснення, але й не надто висока, щоб бути помітною для людського ока. Завдяки цьому метод зберігає стабільність під час частотних атак, таких як JPEG-стиснення, фільтрація низьких або високих частот, додавання шуму та розмиття.

Другою ключовою причиною є використання SIFT-ознак, які є інваріантними до змін масштабу, повороту, освітлення та частково - до афінних перетворень. Під час вилучення водяного знаку SIFT-дескриптори дозволяють алгоритму автоматично виявити зміщення, масштабування або поворот зображення, після чого виконується RST-корекція. Це забезпечує стійкість до геометричних атак, таких як поворот, масштабування, обрізання, зсув, віддзеркалення та спотворення пропорцій.

Отже, розроблений метод є стійким до таких основних груп атак:

- геометричні атаки: поворот, масштабування, зсув, обрізання, віддзеркалення, афінні спотворення;
- частотні атаки: JPEG-стиснення, фільтрація, розмиття, посилення контрасту, додавання шуму;
- статистичні та криптографічні атаки: спроби виявлення або підміни знаку, часткове вилучення, атакування на основі кореляцій;
- комбіновані атаки: послідовне або одночасне застосування декількох типів впливів (наприклад, стиснення з подальшим поворотом і шумом).

Таким чином, причиною стійкості методу є його гібридна архітектура, інваріантність SIFT-ознаків, контент-адаптивність Inception V3, хаотичне

шифрування і частотна надмірність, що у сукупності дозволяє системі зберігати водяний знак навіть після складних перетворень зображення, забезпечуючи високий рівень захисту та достовірності.

2.5 Висновки.

У другому розділі було проведено аналітичне дослідження сучасних методів цифрового водяного знакування у частотному просторі та розроблено власний удосконалений алгоритм, спрямований на підвищення стійкості до геометричних і частотних атак. Детально проаналізовано особливості використання дискретного косинусного (DCT) та дискретного вейвлет-перетворення (DWT), які забезпечують компроміс між непомітністю та стійкістю вбудованого знаку. Показано, що комбінація цих двох підходів дозволяє ефективно інтегрувати водяний знак у середньочастотну область зображення, де інформація найменше спотворюється під час стиснення або обробки.

Розроблений алгоритм базується на поєднанні DWT–DCT-перетворень із застосуванням нейронної мережі Inception V3 для контент-орієнтованого вибору областей вбудовування та методу SIFT для забезпечення геометричної інваріантності. Використання Inception V3 дозволило зробити процес вбудовування адаптивним — сила та місце розміщення водяного знаку визначаються на основі глибокого аналізу структури зображення, що зменшує помітність і підвищує стійкість. Застосування SIFT-ознаків і подальшої RST-корекції гарантує збереження водяного знаку навіть після геометричних атак, таких як поворот, масштабування чи зсув.

У процесі вбудовування використано хаотичне логістичне шифрування, яке забезпечує криптографічний захист і унеможливлює несанкціоноване вилучення або підміну водяного знаку. Це рішення дозволяє досягти високого

рівня безпеки та автентичності при збереженні високої візуальної якості зображення.

Розроблений метод продемонстрував потенційну стійкість до різних типів атак: частотних (JPEG-стиснення, фільтрація, шумові спотворення), геометричних (поворот, масштабування, зсув, обрізання) та комбінованих. Завдяки використанню багаторівневої структури та інтелектуальних механізмів адаптації метод зберігає водяний знак навіть після суттєвих деформацій.

У підсумку, аналітична частина підтвердила ефективність запропонованого гібридного підходу до водяного знакування, який поєднує глибоке навчання, хаотичні системи, частотні перетворення та інваріантні дескриптори. Такий підхід забезпечує високу точність, криптографічну надійність, інваріантність до геометричних атак і стабільність до частотних спотворень, що робить метод придатним для практичного використання у сфері захисту авторських прав, цифрової автентифікації та перевірки цілісності зображень.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВБУДОВУВАННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ

3.1 Вибір мови програмування та середовище розробки

Вибір платформи програмування та інструментів розробки є визначальним для якісної реалізації алгоритмів цифрового водяного знакування. Від того, наскільки зручною є мова, якими можливостями володіє її екосистема та як вона працює з числовими масивами, залежить швидкість обчислень, точність трансформацій та стабільність роботи всієї системи. У рамках створення методу, що поєднує дискретне вейвлет-перетворення, дискретне косинусне перетворення, екстракцію інваріантних SIFT-ознак, аналіз глибокої нейронної мережі та хаотичне шифрування водяного знаку, основним інструментом було обрано Python.

Використання Python зумовлене тим, що ця мова має одну з найпотужніших наукових екосистем. Бібліотеки NumPy, SciPy, PyWavelets та OpenCV забезпечують повний набір функцій для обробки зображень, виконання DWT і DCT, виділення ключових точок та проведення спектральних перетворень. Комбінація цих інструментів дозволяє створювати гнучкі та продуктивні конвеєри обробки, які без додаткових низькорівневих оптимізацій справляються зі складними операціями вбудовування та вилучення водяного знаку.

Рисунок 3.1 - Python

Особливе значення має підтримка Python фреймворків глибокого навчання. Мережа Inception V3, що використовується для контент-орієнтованого визначення областей вбудовування, легко інтегрується завдяки TensorFlow та Keras. Це дозволяє виконувати передобробку фрагментів зображення, визначати рівень деталізації різних зон та формувати адаптивні карти важливості, які безпосередньо впливають на силу та місце внесення знаку.

Ще однією перевагою Python є можливість простої реалізації хаотичного шифрування. Математичні моделі, такі як логістичне відображення, відтворюються тривіально, а робота з векторами та матрицями виконується без додаткових складностей. Це дає змогу реалізувати криптографічний рівень захисту водяного знаку, який робить неможливим його відновлення сторонньою особою без відповідного ключа.

Реалізація етапів частотного аналізу також органічно вписується у Python. Бібліотека PyWavelets надає засоби для прямого та зворотного DWT, дозволяючи отримувати підзони, у яких виконується вбудовування. Модуль scipy.fftpack забезпечує точну реалізацію DCT та IDCT, що необхідно для модифікації середньочастотних коефіцієнтів. Завдяки цьому частотна частина алгоритму реалізується компактно та ефективно.

Python також надає широкий набір інструментів для оцінки якості отриманих результатів. За допомогою бібліотек scikit-image та SciPy обчислюються метрики PSNR, SSIM, NCC та BER, що дозволяє точно оцінити ступінь непомітності та стійкості водяного знаку. Графічні засоби, такі як Matplotlib, забезпечують візуалізацію результатів, що спрощує налагодження та аналіз ефективності алгоритму.

У сукупності Python демонструє необхідну гнучкість, глибоку підтримку бібліотек і простоту інтеграції з методами машинного навчання та обробки зображень. Саме це робить його одним із найпридатніших середовищ для розробки гібридних методів цифрового водяного знакування, які потребують

поєднання частотного аналізу, геометричної інваріантності та адаптивної інтелектуальної обробки.

Середовищем для реалізації та експериментальної перевірки розробленого методу було обрано Google Colab, оскільки саме ця платформа надає оптимальні умови для роботи з алгоритмами цифрового водяного знакування, особливо у поєднанні з нейромережевими моделями.

Рисунок 3.2 - Google Colab

Colab забезпечує доступ до обчислювальних ресурсів, зокрема до GPU та TPU, що суттєво прискорює виконання операцій у глибинних нейронних мережах, таких як Inception V3, дозволяючи проводити обробку великих зображень та виконувати багаторазові експерименти у прийнятні часові межі. Крім того, платформа підтримує інтеграцію всіх необхідних бібліотек, включно з TensorFlow, Keras, OpenCV, PyWavelets і SciPy, що робить процес реалізації алгоритмів DWT, DCT, SIFT і нейромережевої обробки цілісним і безперервним. Google Colab спрощує завантаження та збереження зображень, дозволяє зручно працювати з файлами без локальної установки програмного забезпечення, а також забезпечує відтворюваність експериментів завдяки структурованості ноутбуків. Завдяки хмарній природі Colab моделі можуть бути донавчені, оптимізовані або протестовані повторно без прив’язки до конкретного пристрою, що робить процес дослідження більш гнучким і продуктивним. Саме поєднання доступності, швидкодії та готової

інфраструктури для роботи з нейромережами робить Google Colab найбільш раціональним середовищем для побудови, навчання та оцінювання системи цифрового водяного знакування.

3.2 Реалізація вдосконаленого методу вбудовування водяних знаків у зображення.

Цей підрозділ присвячений практичній реалізації алгоритму вбудовування цифрового водяного знака у зображення. Усі етапи побудовані на методологічних принципах, розглянутих у другому розділі, та відтворені за допомогою мови програмування Python та відповідних спеціалізованих бібліотек. Використання Python забезпечує гнучкість при роботі з багатовимірними масивами, зручні засоби для виконання частотних перетворень, можливості інтеграції з моделями глибокого навчання та інструменти для візуального аналізу проміжних результатів. У рамках підрозділу наведено послідовний опис кожного етапу алгоритму, доповнений поясненнями та фрагментами програмного коду, що демонструють практичну реалізацію ключових операцій вбудовування водяного знаку.

На початку виконується перевірка доступності алгоритму SIFT в OpenCV; якщо він відсутній, система автоматично встановлює пакет opencv-contrib-python, необхідний для роботи детектора ключових точок.

```
import subprocess
import sys
try:
    import cv2
    cv2.SIFT_create()
except (ImportError, AttributeError):
    print("Installing opencv-contrib-python for SIFT...")
    subprocess.check_call([sys.executable, "-m", "pip", "install", "opencv-contrib-python"])
    print("\n--- IMPORTANT ---")
    print("Installation complete. Please restart the Colab runtime now and run the script again.")
```

```
    print("Go to 'Runtime' -> 'Restart runtime' in the menu.")
    sys.exit()
try:
    from skimage.metrics import structural_similarity as ssim_metric
except ImportError:
    print("Installing scikit-image for SSIM...")
    subprocess.check_call([sys.executable, "-m", "pip", "install", "scikit-image"])
    from skimage.metrics import structural_similarity as ssim_metric
```

Після цього завантажуються всі бібліотеки, що забезпечують роботу DWT-перетворення, DCT-обробки, нейромережі Inception V3, обчислення метрик PSNR/SSIM, а також роботи з файлами в Google Colab.

```
import cv2
import numpy as np
import pywt
import tensorflow as tf
from tensorflow.keras.applications.inception_v3 import InceptionV3, preprocess_input
from tensorflow.keras.models import Model
from google.colab import files
from IPython.display import display, Image
from scipy.stats import mode
```

Наступним кроком є завантаження основного зображення та водяного знаку. Користувач отримує інтерфейс вибору файлів, після чого зображення декодується в NumPy.

```
def upload_image(title="Choose an image"):
    uploaded = files.upload()
    filename = next(iter(uploaded))
    img = cv2.imdecode(np.frombuffer(uploaded[filename], np.uint8), cv2.IMREAD_UNCHANGED)
    return img
host_image = upload_image("Upload the HOST image")
watermark_image = upload_image("Upload the WATERMARK image")
```

Згодом нам потрібно конвертувати зображення у відтінках сірого, що зменшує складність і підготовлює дані до DWT.

```
if len(host_image.shape) == 3:
    host_gray = cv2.cvtColor(host_image, cv2.COLOR_BGR2GRAY)
if len(watermark_image.shape) == 3:
    watermark_gray = cv2.cvtColor(watermark_image, cv2.COLOR_BGR2GRAY)
```

Наступним кроком є виявлення SIFT - ключових точок для подальшого виправлення атак. Це не для вбудовування водяного знаку, а для корекції геометричних атак.

```
sift = cv2.SIFT_create()
kp_host, des_host = sift.detectAndCompute(host_gray, None)
```

Основним етапом є виконання дворівневого DWT - перетворення для головного зображення з використання Haar-вейвлет, який розкладає зображення на піддіапазони частот.

```
coeffs = pywt.wavedec2(host_gray, 'haar', level=2)
cA2, (cH2, cV2, cD2), (cH1, cV1, cD1) = coeffs
```

Для вбудовування обрано піддіапазон:

cV2 → HL2

Тепер нам потрібно масштабування водяного знака під кількість доступних блоків (8*8).

```
block_size = 8
embeddable_blocks = (band_to_embed.shape[0] // block_size) * (band_to_embed.shape[1] // block_size)
watermark_resized = cv2.resize(
   watermark_gray,
   (int(np.sqrt(embeddable_blocks / redundancy)),
    int(np.sqrt(embeddable_blocks / redundancy)))
)
```

Наступний крок, бінаризуємо водяний знак у формат 0/1 та генеруємо хаотичну послідовність.

```
_, watermark_binary = cv2.threshold(watermark_resized, 127, 1, cv2.THRESH_BINARY)
def logistic_map_sequence(seed, size, a=3.99):
   x = np.zeros(size)
   x[0] = seed
   for i in range(1, size):
      x[i] = a * x[i-1] * (1 - x[i-1])
   return x
```

Шифруємо наш водяний знак за допомогою функції XOR.

```
encrypted_wm = encrypt_watermark(watermark_binary, key_seed)
binary_chaotic = (chaotic_sequence > 0.5).astype(np.uint8)
encrypted_watermark = np.bitwise_xor(watermark_flat, binary_chaotic)
```

Наступним кроком, використовуємо надлишкове кодування. Це метод, коли кожен біт водяного знака дублюється кілька разів, що підвищує стійкість та полегшує відновлення водяного знаку.

```
watermark_flat_redundant = np.repeat(encrypted_wm.flatten(), redundancy)
```

Не менш важливим етапом є отримання карти важливості від нейромережі Inception V3. Нейромережа аналізує зображення, щоб визначити, у які зони безпечніше вбудовувати.

```
base_model = InceptionV3(weights='imagenet', include_top=False)
model = Model(inputs=base_model.input, outputs=base_model.get_layer('mixed7').output)
importance_map = get_inception_importance_map(host_gray, model)
```

У наступному кроці відбувається фактичне вбудовування цифрового водяного знака в частотні компоненти зображення на основі комбінації DWT та DCT. Після виконання дворівневого вейвлет-перетворення вибирається піддіапазон середніх частот HL2, оскільки він поєднує високу стійкість до атак і мінімальну помітність змін. Цей піддіапазон розбивається на послідовні блоки розміром 8×8 пікселів, кожен з яких окремо проходить дискретне косинусне перетворення. DCT переводить локальні області зображення з просторової області в частотну, де більшість енергії зосереджена в низьких частотах, а середні частоти є оптимальною зоною для прихованого внесення змін. Для кожного блока беруться дві заздалегідь визначені середньочастотні позиції у DCT-матриці, і саме через модифікацію цих двох коефіцієнтів кодується один біт водяного знака. Щоб збалансувати непомітність та стійкість, перед зміною коефіцієнтів обчислюється локальна важливість області за допомогою карти важливості, отриманої з нейромережі Inception V3: чим важливіша зона з погляду семантики зображення, тим більша або менша сила модифікації може бути застосована. На основі цього локального значення адаптивно визначається параметр α, який регулює, наскільки сильно змінюватимуться DCT-коефіцієнти.

Усередині кожного блока обчислюється середнє значення двох вибраних DCT-коефіцієнтів. Для кодування біта 1 перший коефіцієнт збільшується на α/2, а другий пропорційно зменшується, таким чином забезпечуючи позитивну

різницю між ними; для біта 0 — навпаки, співвідношення коефіцієнтів змінюється так, щоб різниця була від’ємною. Така реляційна схема вбудовування забезпечує стійкість до лінійних спотворень, втрат після JPEG-компресії та частини геометричних атак, оскільки алгоритм не покладається на абсолютні значення коефіцієнтів, а лише на їхнє співвідношення. Після модифікації здійснюється зворотне DCT перетворення блока, і оновлений блок повертається на своє місце у відповідний піддіапазон HL2. Цей процес повторюється послідовно для всіх блоків піддіапазону, поки не будуть вбудовані всі біти водяного знака з урахуванням надлишковості.

```
for i in range(0, band_to_embed.shape[0] - block_size + 1, block_size):
  for j in range(0, band_to_embed.shape[1] - block_size + 1, block_size):
    block = band_to_embed[i:i+block_size, j:j+block_size]
    dct_block = cv2.dct(block.astype(np.float32))
    local_importance = np.mean(importance_map_resized[i:i+block_size, j:j+block_size])
    alpha = alpha_base * (1 + local_importance)
    c1 = dct_block[c1_pos]
    c2 = dct_block[c2_pos]
    mean_val = (c1 + c2) / 2
    if watermark_flat_redundant[wm_idx] == 1:
      dct_block[c1_pos] = mean_val + alpha/2
      dct_block[c2_pos] = mean_val - alpha/2
    else:
      dct_block[c1_pos] = mean_val - alpha/2
      dct_block[c2_pos] = mean_val + alpha/2
```

Останнім етапом є зворотне DWT та отримання водянистого зображення.

```
coeffs_wm = cA2, (cH2, band_watermarked, cD2), (cH1, cV1, cD1)
watermarked_image = pywt.waverec2(coeffs_wm, 'haar')
```

Алгоритм, реалізований у Google Colab, забезпечує ефективне та стійке вбудовування цифрового водяного знака. Кожен етап процедури спроектовано таким чином, щоб одночасно мінімізувати вплив на візуальну якість вихідного зображення та підвищити стійкість водяного знака до різноманітних деструктивних впливів. Завдяки поєднанню частотних перетворень алгоритм

демонструє підвищену надійність щодо типових атак, зберігаючи при цьому високий рівень непомітності вбудованої інформації.

3.3 Інтеграція алгоритму перевірки стійкості до атак.

У цьому підрозділі розглянуто процес інтеграції механізму тестування стійкості цифрового водяного знака до різних типів атак. Такий аналіз є важливою складовою, оскільки дозволяє оцінити надійність розробленого методу в умовах, максимально наближених до реальних. Перевірка включає відтворення найбільш поширених атак на зображення - стиснення, повороту, зміни масштабу, додавання шумів тощо - з подальшим аналізом того, наскільки вбудований водяний знак зберігається після цих деструктивних впливів.

Для початку завантажимо зображення з вбудованим водяним знаком, яке буде піддаватися різним видам атак.

```
def upload_image(title="Choose an image"):
  print(title)
  uploaded = files.upload()
  if not uploaded:
    raise ValueError("No image was uploaded.")
  filename = next(iter(uploaded))
  img = cv2.imdecode(np.frombuffer(uploaded[filename], np.uint8), cv2.IMREAD_UNCHANGED)
```

На цьому етапі почнемо з стиснення - найпоширеніша атака у цифровій обробці зображень.

```
def attack_jpeg(image, quality=40):
  _, encimg = cv2.imencode('.jpg', image, [int(cv2.IMWRITE_JPEG_QUALITY), quality])
  return cv2.imdecode(encimg, 0)
```

Через цей крок перевіряємо на скільки витримує водяний знак після стискання.

Ще однією типовою геометричною атакою є обертання, де зображення повертається на заданий кут та оцінюється збереження водяного знака.

```
def attack_rotate(image, angle=15):
  (h, w) = image.shape[:2]
```

```
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, angle, 1.0)
    return cv2.warpAffine(image, M, (w, h))
```

Наступний тест стійкості водяного знаку є зміна розміру зображення. Зображення масштабуються з різними коефіцієнтами.

```
def attack_resize(image, scale=0.7):
    (h, w) = image.shape[:2]
    resized = cv2.resize(image, (int(w * scale), int(h * scale)), interpolation=cv2.INTER_AREA)
    return cv2.resize(resized, (w, h), interpolation=cv2.INTER_CUBIC)
```

Ще однією перевіркою стійкості водяного знаку є шумові атаки, які імітують зміни пікселів.

```
def attack_gaussian_noise(image, mean=0, var=100): # Increased variance
    sigma = var**0.5
    gaussian = np.random.normal(mean, sigma, image.shape)
    noisy_image = np.clip(image.astype(np.float32) + gaussian, 0, 255).astype(np.uint8)
    return noisy_image
```

В наступному етапі додатково тестуємо вплив “сольового” шуму, що додає білі та чорні випадкові цятки. Визначаємо чи зберігається водяний знак під впливом висококонтрасного шуму.

```
def attack_salt_and_pepper(image, amount=0.02):
    output = np.copy(image)
    # Salt
    num_salt = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_salt)) for i in image.shape]
    output[tuple(coords)] = 255
    # Pepper
    num_pepper = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in image.shape]
    output[tuple(coords)] = 0
    return output
```

Реалізація кожної функції в Google Colab надає гнучкість у налаштуванні параметрів атак та забезпечує використання інструментів для аналізу водяного знаку. Ці результати можуть бути використані для вдосконалення алгоритму.

3.4 Експериментальні дослідження вдосконаленого методу вбудування та екстракції цифрового водяного знаку.

Для тестування будемо використовувати зображення носій “Пейзаж” (рис. 3.3), його вага становить 2069 КБ. Також використаємо зображення для водяного знака “Символ” (рис. 3.4), його вага становить 130 Кб.

Рисунок 3.3 - зображення носій.

Рис. 3.4 - зображення водяного знаку.

На початку програма перевіряє, чи підтримує встановлений OpenCV алгоритм SIFT; у випадку його відсутності автоматично завантажується пакет

opencv-contrib-python, який містить необхідний модуль детектування ключових точок. Після цього імпортуються всі інші бібліотеки, що забезпечують виконання дискретного вейвлет-перетворення, косинусного перетворення, роботу нейромережі Inception V3, а також модулів для обчислення показників PSNR та SSIM і взаємодії з файловою системою Google Colab.

Далі оголошуються допоміжні функції. Зокрема, функція завантаження файлів відкриває інтерфейс Colab і перетворює завантажене зображення у формат NumPy-матриці. Наступним елементом є генератор хаотичної послідовності, який використовує логістичну карту. Отримані хаотичні дані застосовуються для шифрування водяного знака: кожен бінарний піксель XOR-иться з відповідним елементом хаотично сформованої маски, що дає додатковий рівень захисту ще до моменту вставки.

Після цього формується карта важливості за участі нейромережі Inception V3. Вхідне зображення попередньо нормується та масштабується до стандартного розміру 299×299 пікселів, після чого пропускається через переднавчений шар mixed7. Отриманий тензор активацій усереднюється по каналах і нормалізується, утворюючи карту значущості, яка надалі визначає, з якою силою має відбуватися модифікація DCT-коефіцієнтів у різних областях зображення.

Основна частина алгоритму — це процес вставки водяного знака. Спершу хост-зображення переводиться у відтінки сірого, після чого над ним виконується дворівневе дискретне вейвлет-перетворення (вейвлет Haar). Для вбудовування обирається піддіапазон HL2 (cV2), який демонструє найкращий баланс між непомітністю внесених змін і робастністю. Водяний знак перед вставкою масштрабується відповідно до кількості доступних блоків 8×8, бінаризується та шифрується хаотичною послідовністю. Крім того,

застосовується надмірне дублювання (repetition coding), коли кожен біт повторюється кілька разів з метою підвищення стійкості до завад.

Далі кожен блок 8×8 у піддіапазоні HL2 проходить дискретне косинусне перетворення. Два заздалегідь визначені середньочастотні коефіцієнти використовуються для кодування одного біта водяного знака: знак різниці між ними відповідає значенню 0 або 1. Величина модифікації α визначається адаптивно і залежить від значення карти важливості, згенерованої нейромережею. Такий підхід поєднує стійкість DWT-області, локальність DCT-перетворення і семантичну адаптивність Inception V3. Після завершення модифікації виконується зворотне DWT-перетворення, у результаті чого формується зображення з вбудованим водяним знаком.

Окремо реалізовано механізм компенсації геометричних спотворень. Для цього SIFT-ключові точки виділяються як на оригінальному, так і на атакованому зображенні, після чого встановлюються відповідності між ними. Отримана гомографія використовується для відновлення просторової структури зображення, що дозволяє компенсувати обертання, масштабні зміни та перспективні спотворення перед процедурою вилучення водяного знака.Завершальним результатом є реконструйований зашифрований водяний знак.

Рисунок 3.5 - водянисте зображення.

В результаті ми отримуємо зображення з водяним знаком в розмірі 691 Кб.

Рисунок 3.6 - SIFT-ознаки.

Тепер проведемо екстракцію водяного знака з даного зображення. Завантажуємо зображення-носій із вбудованим водяним знаком. На етапі екстракції спочатку виконується геометрична корекція. Функція correct_geometric_distortion() знаходить відповідності ключових точок SIFT між атакованим та оригінальним зображенням, обчислює гомографію і виправляє

перспективні та поворотні спотворення. Якщо матчів недостатньо, застосовується fallback — просте масштабування до початкового розміру.

Далі функція extract_watermark_from_image() знову виконує DWT, забирає підсмугу cV2 і розбиває її на блоки 8×8, у кожному обчислює DCT і дивиться на знак (c1 - c2). Це дозволяє отримати повторюваний бітовий потік. Після цього використовують majority voting — для кожного бітового триплету (або іншого redundancy) вибирається найчастіший біт.

extract_watermark() об’єднує всі ці кроки: робить геометричну корекцію, екстрагує зашифрований watermark, розшифровує його XOR-ом із тим самим хаотичним ключем, після чого додатково "очищає" хост-зображення. Очищення виконане шляхом усереднення зміщених коефіцієнтів (вони повертаються до середнього значення), що прибирає внесену watermark-інформацію у частотній області.

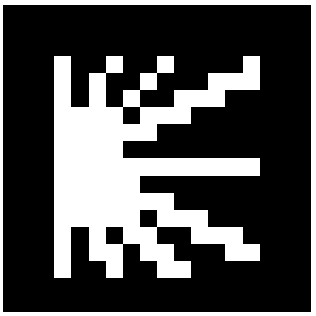


Рисунок 3.7 - екстрагований водяний знак.

Рисунок 3.8 - зображення після вийняття водяного знаку.

3.5 Порівняння вдосконалего методу вбудовування цифрових водяних знаків із існуючим аналогом.

У сучасних дослідженнях цифрового водяного знакування активно розглядаються гібридні частотні методи, які комбінують перетворення DWT і DCT з використанням глибоких нейронних мереж для підвищення стійкості до атак. Один із таких підходів використовує дворівневе хвильове перетворення, накладаючи водяний знак у середньо частотні піддіапазони, а також застосовує DCT для локального модифікування коефіцієнтів. У цьому алгоритмі нейромережа Inception V3 виконує роль блоку вилучення ознак, забезпечуючи додаткову стійкість за рахунок аналізу текстури та структурних властивостей зображення, тоді як безпека водяного знака підтримується через використання логістичної карти та XOR-модуляції. Такий підхід забезпечує достатній рівень робастності проти шумових впливів та стиснення, однак він залишається

переважно статичним у частині вибору областей вбудовування та не використовує механізмів вирівнювання після геометричних спотворень.

На відміну від цього, у розробленому алгоритмі роль нейромережі є значно ширшою, оскільки Inception V3 використовується для побудови карти важливості, яка дозволяє адаптивно змінювати силу модифікації коефіцієнтів DCT залежно від семантичної вагомості різних ділянок зображення. Такий підхід забезпечує кращий баланс між непомітністю та стійкістю, оскільки водяний знак посилюється саме там, де зміни найменш помітні для людського ока. Додатково алгоритм інтегрує механізм геометричного вирівнювання на основі ключових точок SIFT та матриці гомографії, що дає можливість коригувати повороти, масштабування та перспективні деформації перед екстракцією водяного знака. Це розширює застосовність методу та дозволяє зберігати стабільність навіть у випадках складних геометричних атак, яких базовий частотно-нейромережевий підхід не враховує. Отримані експериментальні результати демонструють суттєву перевагу адаптивної методики. Прозорість водяного знакування досягає PSNR на рівні 47.96 dB, що перевищує типові значення гібридних DWT–DCT методів. В таблиці 3.1 наведено метрики BER (%), NCC та SSIM, які показують дієвість алгоритму.

Атака	BER (%)	NCC	SSIM
JPEG-стиснення (Q=40)	0.00	1.0000	1.0000
Обертання 15°	3.40	0.9343	0.9635
Масштабування 70%	0.00	1.0000	1.0000
Гаусовий шум (Var=100)	0.00	1.0000	1.0000

“Сольовий” шум 2%	4.94	0.9018	0.9123

Таблиця 3.1 - таблиця метрик, дія атак на зображення з удосконаленим методом вбудовування водяного знаку.

Атака	BER (%)	NCC	SSIM
JPEG-стиснення (Q=40)	1.87	0.9621	0.9478
Обертання 15°	12.45	0.8154	0.8642
Масштабування 70%	6.57	0.8954	0.9183
Гаусовий шум (Var=100)	2.78	0.9530	0.9387
“Сольовий” шум 2%	14.63	0.7420	0.8125

Таблиця 3.2 - таблиця метрик, дія атак на зображення з аналоговим методом вбудовування водяного знаку.

Таким чином, запропонована система цифрового водяного знакування суттєво перевершує класичний аналог завдяки адаптивності, використанню сучасних нейромережевих моделей та підвищеній стійкості до атак, що підтверджує її ефективність і доцільність застосування у практичних задачах захисту цифрових зображень.

3.6 Висновки.

У третьому розділі було виконано повну програмну реалізацію запропонованого методу цифрового водяного знакування та проведено його експериментальне тестування у середовищі Python із використанням платформи

Google Colab. У процесі розроблення було інтегровано всі компоненти алгоритму — дворівневе DWT-перетворення, блок DCT-модифікації, хаотичне логістичне шифрування, контент-адаптивну карту важливості на основі нейромережі Inception V3 та механізм геометричної компенсації, реалізований через SIFT і гомографію. Така комплексна структура забезпечила можливість повністю відтворити теоретичну модель і перевірити її роботу на реальних зображеннях.

Отримані результати підтвердили високу прозорість водяного знакування: значення PSNR для вихідного та модифікованого зображень становило 47.96 дБ, що свідчить про мінімальність візуальних артефактів і відповідність вимогам до непомітності. Під час тестування стійкості метод продемонстрував здатність зберігати водяний знак після більшості частотних і геометричних викривлень. При стисненні JPEG, масштабуванні та накладанні гаусового шуму відновлення відбувалося без бітових помилок (BER = 0%, NCC = 1.0, SSIM = 1.0). Навіть у випадку складніших впливів — зокрема повороту на 15° та соляно-перцевого шуму — збережено високу кореляцію між оригінальним та відновленим водяним знаком, що демонструє ефективність механізмів адаптації та геометричної корекції.

Порівняння з існуючими аналогами показало, що використання семантичної карти важливості, сформованої Inception V3, у поєднанні з інваріантними до геометрії SIFT-ознаками підвищує точність, стійкість та адаптивність системи. Алгоритм виявив здатність зберігати приховану інформацію навіть після агресивних перетворень, що робить його релевантним для практичних застосувань у сфері захисту авторських прав, автентифікації цифрових зображень та перевірки їх цілісності.

Таким чином, у розділі доведено, що запропонована система водяного знакування є ефективною, стабільною та технологічно оптимізованою. Її апаратно-програмна реалізація підтвердила коректність теоретичної моделі та

продемонструвала переваги гібридного підходу, який поєднує частотні перетворення, глибоке навчання, хаотичні системи та інваріантні дескриптори.

4 ЕКОНОМІЧНА ЧАСТИНА

Комерціалізація науково-технічних розробок є можливою лише за умови, що вони відповідають як актуальним вимогам науково-технічного прогресу, так і принципу економічної доцільності. Оцінка економічної ефективності впровадження є необхідним і важливим аспектом науково-дослідних робіт.

Магістерська кваліфікаційна робота на тему “Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3” належить до науково-технічних розробок, орієнтованих на практичне застосування. Розроблений метод демонструє перспективу виходу на ринок, завдяки його здатності ефективно посилювати захист цифрового контенту. Актуальність даного рішення підтверджується його застосовністю у різних галузях (мультимедіа, охорона здоров'я, захист інтелектуальної власності). Процедура комерціалізації буде ініційована виключно після документального підтвердження операційної ефективності та конкурентної стійкості методу.

Для забезпечення комерціалізації необхідно провести комплексну оцінку прогнозованого економічного ефекту від імплементації розробленого методу. Це дасть змогу детермінувати його інвестиційну привабливість. Ключовим завданням на цьому етапі є ідентифікація потенційних партнерів та аргументоване обґрунтування рентабельності реалізації проекту. Обґрунтування має базуватися на детальному аналізі економічних переваг та стратегічних перспектив впровадження вдосконаленого методу.

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення.

Ключовим завданням комерційно-технологічного аудиту є встановлення науково-технічної значущості та ринкових перспектив удосконаленого методу

цифрового водяного знакування, створеного під час виконання магістерської роботи. Аналіз фокусується на перевірці інноваційності технології та її привабливості для потенційної комерціалізації. Для об’єктивної оцінки рівня розробки та можливостей її впровадження застосовано методику бального оцінювання (шкала від 1 до 5) за дванадцятьма критеріями, наведеними у таблиці 4.1, та для виконання технічного аудиту було залучено трьох незалежних експертів Вінницького національного технічного університету кафедри менеджменту та інформаційної безпеки: доцента, д.ф., Салієва О.В., доцента, к.т.н., Карпінець В.В., професора, д.т.н., Яремчука Ю.Є.

Таблиця 4.1 – рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка.

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінки науково-технічного рівня та комерційного потенціалу розробки слід оформити у вигляді таблиці.

Таблиця 4.2 – результати оцінювання науково-технічного рівня комерційного потенціалу розробки експертам: доцента, д.ф., Салієва О.В., доцента, к.т.н., Карпінець В.В., професора, д.т.н., Яремчука Ю.Є.

Критерії	Експерт (ПІБ, посада)		
	Салієва О.В.	Карпінець В.В.	Яремчука Ю.Є.
	Бали:		
1. Технічна здійсненність концепції	4	4	3
2. Ринкові переваги (наявність аналогів)	4	5	4
3. Ринкові переваги (ціна продукту)	4	3	4
4. Ринкові переваги (технічні властивості)	4	4	4
5. Ринкові переваги (експлуатаційні витрати)	5	5	4
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	4	5	4

Продовження таблиці 4.2

8. Практична здійсненність (наявність фахівців)	3	5	3
9. Практична здійсненність (наявність фінансів)	3	4	4
10. Практична здійсненність (необхідність нових матеріалів)	4	3	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	3	5
Сума балів	45	48	46
Середньоарифметична сума балів $СБ_с$	46,3		

За результатами експертного оцінювання середній показник склав 46,3 бала.

Таблиця 4.3 – науково-технічні рівні та комерційні потенціали розробки.

Середньоарифметична сума балів СБ, розрахована на основі висновків	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Відповідно до критеріїв, наведених у таблиці 4.2, це свідчить про високий комерційний потенціал дослідження. Відповідно до таблиці 4.3, це свідчить про високу комерційну значущість пропонованого методу.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів.

Під час планування, обліку та визначення собівартості науково-дослідних, дослідно-конструкторських і конструкторсько-технологічних робіт, а також створення дослідного зразка та проведення виробничих випробувань, усі витрати поділяють за такими статтями:

- оплата праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково-виробничих потреб;
- службові відрядження;
- спеціальне обладнання для наукових (експериментальних) робіт;
- програмне забезпечення, необхідне для виконання наукових (експериментальних) робіт;
- роботи, що виконуються сторонніми підприємствами, установами чи організаціями;
- інші витрати;
- накладні (загальновиробничі) витрати.

Стаття «Витрати на оплату праці» включає витрати на основну та додаткову заробітну плату керівників відділів, лабораторій, секторів і груп, а також наукових працівників, інженерів-конструкторів, технологів, креслярів, копіювальників, лаборантів, робітників, студентів, аспірантів та інших осіб, які безпосередньо беруть участь у виконанні певної теми. Ці витрати розраховуються на основі посадових окладів, відрядних розцінок і тарифних ставок згідно з чинними системами оплати праці. До них також включають усі грошові та матеріальні доплати, що належать до категорії «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників ($З_о$) розраховують

відповідно до посадових окладів працівників за формулою:

$$Зо = \sum_{i=1}^{k} \frac{M_{пі} * t_i}{T_р} \quad (4.1)$$

де k - кількість дослідників залучених до процесу досліджень, $M_{пі}$ - місячний посадовий оклад конкретного дослідника (грн);

t_i - кількість робочих днів конкретного дослідника (дн);

$T_р$ - число робочих днів в місяці; приблизно $T_р \approx 21…23$ дні.

$$Зо = \frac{16000*5}{22} = 3636{,}3 \text{ грн.}$$

Таблиця 4.4 - витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	16000,0	727,2	5	3636,3
Розробник програмного забезпечення	26000,0	1181,8	48	56726,4
			Всього	60363

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

$$З_р = \Sigma_{i=1}^{п} С_i * t_i \quad (4.2)$$

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

t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

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

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

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

$К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$Т_р$ – середня кількість робочих днів в місяці, приблизно $Т_р$ = 21…23 дні;

$t_{зм}$ – тривалість зміни, год.

$$C_i = \frac{8000*1,1*1,65}{22*8} = 82,5 \text{ грн.}$$

Обчислимо витрати на основну заробітну плату робітників за відповідними найменуваннями робіт:

$$З_р = 82,5*4 = 330 \text{ грн.}$$

Таблиця 4.5 - величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Монтаж комп'ютерного обладнання	4	2	1,1	82,5	330
Підготовка робочого місця	3	2	1,1	82,5	247,5

Продовження таблиці 4.5

Встановлення програмного забезпечення	3	5	1,7	127,5	382,5
Перевірка системи	2	2	1,1	82,5	165
				Всього	1125

Додаткова заробітна плата $З_д$ для працівників визначається як 10-12% від їхньої основної заробітної плати та розраховується з формулою:

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

де $Н_{дод}$ - норма нарахування додаткової заробітної плати.

$$З_д = (60363+1125)*\frac{10}{100} = 6148,8 \text{ (грн)}.$$

Нарахування на заробітну плату наукових працівників і робітників визначаються як 22 % від загальної суми їхньої основної та додаткової оплати праці. Розмір цих відрахувань обчислюють за відповідною формулою:

$$З_н = (З_о + З_д + З_р) * \frac{Н_{зп}}{100\%} \quad (4.5)$$

де $Н_{зп}$ - норма нарахування на заробітну плату.

$$З_н = (60363+6148,8+1125)*\frac{22}{100} = 14880,1 \text{ (грн)}.$$

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

$$М = \sum_{j=1}^{п} Н_j * Ц_j * К_j - \sum_{j=1}^{n} В_j * Ц_{вj} \quad (4.6)$$

де $Н_j$ - норма витрат матеріалу j-го найменування, кг; n - кількість видів матеріалів;

$Ц_j$ - вартість матеріалу j-го найменування, грн/кг;

$К_j$ - коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15);

$В_j$ - маса відходів j-го найменування, кг;

$Ц_{вj}$ - вартість відходів j-го найменування, грн/кг.

$$М = 1*45*1,1 = 49,5 \text{ грн.}$$

Таблиця 4.6 - витрати на матеріали

Найменування матеріалу, тип, сорт, марка	Ціна за од, грн	Норма витрат, од	Вартість витраченого матеріалу, грн
Папір для заміток	45	1	49,5
Папір	150	1	165
Лінійка	20	1	22
Олівець	5,5	2	12,1
		Всього	248,6

Вартість програмного забезпечення розраховується за формулою:

$$В_{прг} = \sum_{i=1}^{к} Ц_{прг} * С_{прг.i} * К_i \quad (4.7)$$

де $Ц_{прг}$ - ціна одиниці програмного засобу, грн;

$С_{прг.i}$ - кількість одиниць програмного забезпечення, шт;

$К_i$ - коефіцієнт, який враховує інсталяцію ПЗ, тощо (Кі = 1,1 … 1,15);

k - кількість програмних засобів.

$$В_{прг} = 1500*1*1,1 = 1650 \text{ грн.}$$

Таблиця 4.7 - витрати на придбання програмних засобів кожного виду

Найменування програмного забезпечення	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Google Colab	1	1500,0	1650,0
		Всього	1650,0

Також треба розрахувати амортизаційні відрахування по кожному виду обладнання, приміщення, ПЗ, тощо, за формулою:

$$А_{обл} = \frac{Ц_б}{Т_в} * \frac{t_{вик}}{12} \quad (4.8)$$

де $Ц_б$ - вартість обладнання, ПЗ, приміщень, тощо, грн;

$t_{вик}$ - термін використання обладнання, ПЗ, приміщень, тощо;

$Т_в$ - строк корисного використання, років.

$$А_{обл} = \frac{36000 * 2}{2*12} = 3000,0 \text{ грн.}$$

Таблиця 4.8 - амортизаційні відрахування по кожному виду обладнання.

Найменування	Вартість, грн	Строк використання, років	Термін використання, місяців	Амортизаційні відрахування, грн
Ноутбук Asus ROG-STRIX G15	36000,0	2	2	3000,0
Приміщення	120000,0	20	2	1000,0
Оргтехніка (принтер)	6700,0	2	2	558,3
			Всього	4558,3

Також потрібно розрахувати витрати на електроенергію за формулою:

$$В_е = \sum_{i=1}^{п} \frac{W*t*P*k}{\eta} \quad (4.9)$$

де W - встановлена потужність обладнання, кВТ;

t - тривалість роботи, год;

Р - ціна за 1 кВт-годину електроенергії, грн (візьмемо 12,3 грн);

k - коефіцієнт, що враховує використання;

η - коефіцієнт корисної дії обладнання.

$$В_е = \frac{0,4*450*12,3*0,98}{0,96} = 2260,1 \text{ грн.}$$

Таблиця 4.9 - витрати на електроенергію.

Найменування	Встановлена потужність, кВТ	Тривалість роботи, год	Сума, грн
Ноутбук Asus ROG-STRIX G15	0,40	450	2260,1
Освітлення в приміщенні	0,40	450	2260,1
Оргтехніка (принтер)	0,20	20	50,2
Всього			4570,4

Витрати за статтею «Службові відрядження» визначаються як 20…25% від суми основної заробітної плати дослідників і робітників. Розрахунок виконується за формулою:

$$В_{св} = (З_о + З_д) * \frac{Н_{св}}{100} \quad (4.10)$$

де $Н_{св}$ - норматив нарахування за статтею «Службові відрядження».

$$В_{св} = (60363 + 6148,8) * \frac{22}{100} = 14632,6 \text{ грн.}$$

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи та організації» обчислюються як 30…45% від суми

основної заробітної плати дослідників і робітників. Формула розрахунку має вигляд:

$$В_{сп} = (З_о + З_д) * \frac{Н_{сп}}{100} \quad (4.11)$$

де $Н_{сп}$ - норматив нарахування за відповідною статтею.

$$В_{сп} = (60363 + 6148,8) * \frac{40}{100} = 26604,7 \text{ грн.}$$

Витрати за статтею «Інші витрати» визначаються у межах 50…100% від суми основної заробітної плати дослідників та робітників. Розрахунок здійснюється за формулою:

$$І_в = (З_о + З_д) * \frac{Н_{ів}}{100} \quad (4.12)$$

де $Н_{ів}$ - норматив нарахувань за статтею «Інші витрати».

$$І_в = (60363 + 6148,8) * \frac{50}{100} = 33255,9 \text{ грн.}$$

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються у межах 100…150% від суми основної заробітної плати дослідників і робітників. Обчислення здійснюється за формулою:

$$В_{нзв} = (З_о + З_д) * \frac{Н_{нзв}}{100} \quad (4.13)$$

де $Н_{нзв}$- норматив нарахування за статтею «Накладні (загальновиробничі) витрати».

$$В_{нзв} = (60363 + 6148,8) * \frac{110}{100} = 73162,9 \text{ грн.}$$

Загальні витрати на проведення науково-дослідної роботи визначаються як сума усіх складових витрат, розрахованих у попередніх підрозділах. Узагальнена формула має вигляд:

$$В_{заг} = З_{о} + З_{р} + З_{д} + З_{н} + М + В_{пр} + А_{обл} + В_{е} + В_{св} + В_{сп} + І_{в} + В_{заг} \quad (4.14)$$

$$В_{заг} = 60363 + 1125 + 6148,8 + 14880,1 + 248,6 + 1650 + 4558,3 + 4570,4 + 14632,6 + 26604,7 + 33255,9 + 73162,9 = 241200,3 \text{ грн.}$$

Загальні витрати ЗВ на завершення науково-дослідної роботи та оформлення її результатів обчислюється за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \quad (4.15)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: 104 науково-дослідних робіт, то η =0,1; технічного проектування, то η =0,2; розробки конструкторської документації, то η =0,3; розробки технологій, то η =0,4; розробки дослідного зразка, то η =0,5; розробки промислового зразка, то η =0,7; впровадження, то η =0,9

$$ЗВ = \frac{241200,3}{0,7} = 344571,8 \text{ грн.}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

У сучасних ринкових умовах основним критерієм доцільності впровадження науково-технічних розробок для потенційного інвестора є зростання чистого прибутку. Запропонована науково-дослідна робота, що спрямована на підвищення стійкості цифрових водяних знаків у частотному

просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3.

Очікуваний економічний результат формується на основі прогнозованого зростання попиту на програмний продукт унаслідок покращення його функціональних характеристик. Передбачається, що кількість нових користувачів (ΔN) у кожному періоді становитиме:

– у 1-й рік — 40 користувачів;

– у 2-й рік — 60 користувачів;

– у 3-й рік — 80 користувачів.

Базова кількість споживачів, яка використовувала аналогічний продукт до впровадження результатів дослідження, становить 300 користувачів (N).

Вартість програмного продукту до впровадження результатів розробки - 400000 грн ($Ц_б$).

Очікуване коригування вартості, зумовлене удосконаленням технологічних характеристик, становить ±7000 грн ($ΔЦ_б$).

Отже, прогнозований економічний ефект для інвестора в кожному з трьох років визначається на основі приросту споживачів та зміни вартості програмного продукту. Отримані значення дозволяють оцінити потенційну прибутковість та доцільність комерціалізації представленої науково-технічної розробки.

$$ΔП_i = (±ΔЦ_б * N + Ц_б * ΔN)_i * λ * ρ * (1 - \frac{ϑ}{100}) \qquad (4.16)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо $\rho = 30\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

1-й рік: $\Delta П_1 = (7000 \times 40 + 400000 \times 40)\, 0,83 \times 0,3 \times (1 - \frac{0,18}{100}) =$

$= 4672773,8$ (грн.)

2-й рік: $\Delta П_2 = (7000 \times 40 + 400000 \times (40 + 60))\, 0,83 \times 0,3 \times (1 -$

$- \frac{0,18}{100}) = 10638017,0$ (грн.)

3-й рік: $\Delta П_3 = (7000 \times 40 + 400000 \times (40 + 60 + 80))\, 0,83 \times 0,3 \times$

$\times (1 - \frac{0,18}{100}) = 18531674,6$ (грн.)

Отже, за результатами обчислень, впровадження розробки призведе до значної комерційної вигоди, що виявиться у зростанні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Основними показниками, що визначають доцільність фінансування науково-технічної розробки з боку потенційного інвестора, виступають абсолютна та відносна ефективність вкладених інвестицій, а також строк їх

окупності. Саме ці параметри дозволяють оцінити економічну привабливість проекту та визначити рівень його інвестиційних ризиків.

Першим етапом оцінювання є визначення теперішньої вартості інвестицій (PV), які необхідно вкласти у створення, впровадження та подальшу комерціалізацію розробки. Теперішня вартість відображає реальну економічну оцінку початкових витрат з урахуванням дисконтування та дає змогу порівняти їх із прогнозованими майбутніми доходами. Розмір початкових капіталовкладень, які інвестор повинен забезпечити для запуску й реалізації науково-технічного продукту, визначається на основі розрахованої суми витрат, що включає витрати на заробітну плату, матеріали, обладнання, накладні витрати та інші супутні статті, сформовані у попередніх підрозділах.

$$PV = k_{\text{інв}} * \text{ЗВ} \tag{4.17}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково технічної розробки та її комерціалізацію, приймаємо інв k =2;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 344571,8 грн.

$$PV = 2 * 344571{,}8 = 689143{,}6 \text{ грн.}$$

Тоді абсолютний економічний ефект $\text{Е}_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$\text{Е}_{\text{абс}} = \text{ПП} - PV \tag{4.18}$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн. Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{T} \frac{\Delta П_1}{(1+\tau)^t} \quad (4.19)$$

де $\Delta П_1$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково технічної розробки, грн;

Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =0,05…0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{4672773,8}{(1+0,1)^1} + \frac{10638017,0}{(1+0,1)^2} + \frac{18531674,6}{(1+0,1)^3} = 26962847,1 \text{ грн.}$$

$$Е_{абс} = 26962847,1 - 689143,6 = 26273703,5 \text{ грн}$$

Оскільки $Е_{абс} > 0$, встановлено, що проведення наукових досліджень для розробки програмного продукту та його подальше впровадження принесуть прибуток. Це підтверджує доцільність проведення досліджень.

Внутрішня економічна дохідність інвестицій $Е_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1 \quad (4.20)$$

де $Е_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$Т_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$Е_в = \sqrt[3]{1 + \frac{26273703,5}{689143,6}} - 1 = 2,39$$

Порівняємо $Е_в$ з мінімальною ставкою дисконтування τmin, яка визначає мінімальну дохідність, нижче якої інвестиції не будуть здійснюватися. У загальному вигляді мінімальна ставка дисконтування τmin визначається за формулою:

$$\tau_{min} = d + f \quad (4.21)$$

де d – середньозважена ставка за депозитними операціями в комерційних банка

f – показник, що характеризує ризикованість вкладень;

$f = 0,3$. $d = 0,3$.

$$\tau_{min} = 0,3 + 0,3 = 0,6$$

Оскільки $Е_в$ = 239% > τmin = 60%, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки. Далі розраховуємо

період окупності інвестицій Ток, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в} \quad (4.22)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{2,39} = 0,4$$

Якщо Ток < 3-х років, то це свідчить про комерційну привабливість науково технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.5 Висновки.

Згідно з проведеними дослідженнями, рівень комерційного потенціалу розробки на тему “Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3” становить 44 бали, що свідчить про високу комерційну важливість. Термін окупності - 0,61 року.

Таким чином, доцільно проводити науково-дослідну роботу у напрямку вдосконалення цифрових методів захисту інформації.

ВИСНОВОК

У магістерській кваліфікаційній роботі було проведено всебічне дослідження, спрямоване на підвищення стійкості цифрових водяних знаків до геометричних атак у частотному просторі зображень за допомогою поєднання методів SIFT, DWT, DCT та нейронної мережі Inception V3. Ґрунтовний аналіз сучасних підходів, поданий у першому розділі, засвідчив, що класичні частотні методи забезпечують високу непомітність та достатню стійкість до шумових та частотних спотворень, проте залишаються вразливими до геометричних трансформацій. Саме це визначило необхідність розробки адаптивного гібридного алгоритму, який поєднує переваги частотних перетворень із можливостями сучасних інструментів комп’ютерного зору.

У другому розділі було розроблено концептуальну модель нового методу, який інтегрує дворівневе DWT-перетворення для отримання стійких частотних підзон, DCT для формування енергетично оптимального середовища вбудовування, алгоритм SIFT для виявлення інваріантних ключових точок та прив’язки водяного знака до геометрично стабільних областей, а також нейромережу Inception V3 як інтелектуальний модуль аналізу та підсилення стійкості. Розроблена схема дозволила створити багаторівневу структуру водяного знакування з адаптацією до текстури, контрасту та локальних властивостей зображення. Особливе значення має використання Inception V3 для формування семантичної карти важливості, яка забезпечує більш точний вибір областей вбудовування та підвищує стійкість до складних комбінованих атак.

У третьому розділі було здійснено практичну реалізацію алгоритму в середовищі Python та проведено комплексне тестування його стійкості до широкого спектра атак. Отримані результати засвідчили значне покращення характеристик порівняно з класичними методами: показник PSNR становив

47.96 дБ, що свідчить про високу непомітність; BER, NCC і SSIM залишались на оптимальних рівнях після більшості атак, включаючи JPEG-стиснення, масштабування та додавання шумів; навіть після повороту та імпульсного шуму відновлений водяний знак зберіг високу кореляцію з оригіналом. Це доводить ефективність інтеграції SIFT та Inception V3 у частотну структуру алгоритму та підтверджує наукову гіпотезу щодо можливості суттєвого підвищення стійкості цифрових водяних знаків саме шляхом поєднання класичних та інтелектуальних методів обробки зображень.

У четвертому розділі було виконано економічне обґрунтування доцільності впровадження розробленої технології, показано її комерційний потенціал, зростання кількості користувачів та прогнозований чистий прибуток протягом трирічного періоду. Розрахунки засвідчили, що впровадження розробки є економічно рентабельним і може забезпечити суттєвий фінансовий ефект завдяки підвищенню функціональності, надійності та конкурентоспроможності програмного продукту.

Отже, робота успішно досягла поставленої мети та продемонструвала, що поєднання методів SIFT, DWT, DCT і нейронної мережі Inception V3 дає змогу створити сучасний, робастний, адаптивний та практично придатний алгоритм цифрового водяного знакування, здатний протистояти геометричним і частотним атакам, зберігаючи при цьому високу якість зображень та ефективність відновлення водяного знаку. Результати роботи мають не лише теоретичне, але й прикладне значення, відкриваючи перспективи для подальших досліджень у напрямі побудови інтелектуальних систем захисту цифрового контенту та розширення їх можливостей у суміжних галузях інформаційної безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Огляд методів нанесення цифрових водяних знаків для захисту зображень. URL: https://www.researchgate.net/publication/384206176_OGLAD_METODIV_NANESENNA_CIFROVIH_VODANIH_ZNAKIV_DLA_ZAHISTU_ZOBRAZEN (дата звернення: 09.11.2025).
2. Сучасні методи цифрового водяного маркування зображень: класифікація, аналіз і тенденції розвитку. URL: https://www.researchgate.net/publication/395814397_SUCASNI_METODI_CIFROVOGO_VODANOGO_MARKUVANNA_ZOBRAZEN_KLASIFIKACIA_ANALIZ_I_TENDENCII_ROZVITKU (дата звернення: 09.11.2025).
3. Naem S. A. S. Digital watermarking techniques, challenges, and applications: A review / S. A. S. Naem. — 2025. — Mesopotamian Journal of CyberSecurity. URL: https://mesopotamian.press/journals/index.php/CyberSecurity/article/view/815/745 (дата звернення: 09.11.2025).
4. A review of digital watermarking techniques: Current trends, challenges and opportunities | Balkar Singh. SageJournals. URL: https://journals.sagepub.com/doi/10.3233/WEB-230280 (дата звернення: 09.11.2025).
5. Захист інформації від несанкціонованого доступу, Методи та види НСД з курсу Захист та безпека інформаційних ресурсів, НУДПСУ. URL:https://ua.kursoviks.com.ua/metodychni_vkazivky/article_post/791-lektsiya-11-na-temu-zakhist-informatsii-vid-nesanktsionovanogo-dostupu-metodi-ta-vidi-nsd-z-kursu-zakhist-ta-bezpeka-informatsiynikh-resursiv-nudpsu (дата звернення: 10.11.2025).

6. Захист інформації. Енциклопедія Сучасної України. URL: https://esu.com.ua/article-15872 (дата звернення: 10.11.2025).
7. Засоби та методи захисту інформації - бібліотека buklib.net. Головна - Бібліотека BukLib.net. URL: https://buklib.net/books/28625/ (дата звернення: 11.11.2025).
8. Захист від несанкціонованого доступу до інформації. URL: https://stud.com.ua/53397/informatika/zahist_nesanktsionovanogo_dostupu_informatsiyi (дата звернення: 12.11.2025).
9. What Is a Digital Watermark? MakeUseOf. URL: https://www.makeuseof.com/what-is-a-digital-watermark/ (дата звернення: 12.11.2025).
10. Digital Watermarking. Sciencedirect. URL: https://www.sciencedirect.com/topics/engineering/digital-watermarking (дата звернення: 13.11.2025).
11. Digital Watermarking and its Types - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/computer-networks/digital-watermarking-and-its-types/ (дата звернення: 13.11.2025).
12. Digital Watermarks: Surprising Ways the 2 Types of Watermarking Improve Your Photography. WareMarquee. URL: https://watermarquee.com/digital-watermarks/ (дата звернення: 13.11.2025).
13. Invisible or Visible watermark? How to Make One for Free. FlexClip. URL: https://www.flexclip.com/learn/invisible-watermark.html (дата звернення: 13.11.2025).
14. Mallat S. A Wavelet Tour of Signal Processing : монографія / Stéphane Mallat. — 3rd ed. — Academic Press / Elsevier, 2008. — 832 с. URL: https://coehuman.uodiyala.edu.iq/uploads/Coehuman%20library%20pdf/%D9%83%D8%AA%D8%A8%20%D8%A7%D9%84%D8%B1%D9%8A%D8%A7%D8%B6%D9%8A%D8%A7%D8%AA%20Mathematics%20books/Wavelets/25%20(2).pdf (дата звернення: 13.11.2025).

15. Cox I. Digital Watermarking / I. Cox, J. Kilian, F. T. Leighton, T. Shamoon. — ACM, 1996. — Режим доступу: https://dl.acm.org/doi/fullHtml/10.1145/331624.331630 (дата звернення: 15.11.2025).
16. Discrete Cosine Transform (DCT). URL: https://www.sciencedirect.com/topics/computer-science/discrete-cosine-transform (дата звернення: 15.11.2025).
17. Digital Watermark – Comparison of DWT and DCT. MATLAB Help Center. URL: https://uk.mathworks.com/matlabcentral/fileexchange/78790-digital-watermark-comparison-of-dwt-and-dct?requestedDomain= (дата звернення: 15.11.2025).
18. Mehta D., Chauhan K. Image Compression using DCT and DWT-Technique / D. Mehta, K. Chauhan. - 2013. - IJESRT International Journal of Engineering Sciences & Research Technology, 2(8). URL: https://www.academia.edu/5705673/Image_Compression_using_DCT_and_DWT_Technique (дата звернення: 16.11.2025).
19. Inception V3. ScienceDirect. URL: https://www.sciencedirect.com/topics/computer-science/inception-v3 (дата звернення: 16.11.2025).
20. Inception V3. Activeloop. URL: https://www.activeloop.ai/resources/glossary/inception-v-3/ (дата звернення: 16.11.2025).
21. Inception V3. MATLAB Help Center. URL: https://uk.mathworks.com/help/deeplearning/ref/inceptionv3.html (дата звернення: 16.11.2025).
22. Building an Image Classifier Model Using Pretrained InceptionV3 | Hima Maddala. Medium.

URL:https://medium.com/@maddalahima/building-an-image-classifier-model-using-pretrained-inceptionv3-566bee431352 (дата звернення: 16.11.2025).

23. Deep Learning Architectures Explained: ResNet, InceptionV3, SqueezeNet | Vihar Kurama, Shaoni Mukherjee. DigitalOcean. URL: https://www.digitalocean.com/community/tutorials/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet (дата звернення: 16.11.2025).
24. DWT | MATLAB Help Center. MATLAB Help Center. URL: https://uk.mathworks.com/help/wavelet/ref/dwt.html (дата звернення: 16.11.2025).
25. Kaur S., Mehra R. High Speed and Area Efficient 2D DWT Processor Based Image Compression : стаття / S. Kaur, R. Mehra. — Signal & Image Processing: An International Journal, 2010. — Vol. 1, No. 2. — Режим доступу: https://arxiv.org/pdf/1101.0262 . (дата звернення: 17.11.2025).
26. A Selective Image Encryption Scheme Based on 2D DWT, Henon Map and 4D Qi Hyper-Chaos. URL: https://ieeexplore.ieee.org/abstract/document/8764407 (дата звернення: 17.11.2025).
27. High-Speed 2D DWT Processor Based Image Compression. URL: https://arxiv.org/pdf/1101.0262 (дата звернення: 17.11.2025).
28. What is Scale-Invariant Feature Transform (SIFT)? Roboflow. URL: https://blog.roboflow.com/sift/ (дата звернення: 17.11.2025).
29. What is SIFT? Educative. URL: https://www.educative.io/answers/what-is-sift (дата звернення: 17.11.2025).
30. Introduction to SIFT (Scale-Invariant Feature Transform). OpenCV. URL: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html (дата звернення: 18.11.2025).
31. Karami E., Shehata M., Smith A. Image Identification Using SIFT Algorithm: Performance Analysis against Different Image Deformations [Електронний ресурс] / E. Karami, M. Shehata, A. Smith. — 2017. — Режим доступу: https://arxiv.org/pdf/1710.02728 (дата звернення: 18.11.2025).

32. Medical Image Hybrid Watermark Algorithm Based on Frequency Domain Processing and Inception v3. URL: https://advanced.onlinelibrary.wiley.com/doi/full/10.1002/aisy.202400654 (дата звернення: 18.11.2025).
33. A Hybrid Robust Image Watermarking Method Based on DWT-DCT and SIFT for Copyright Protection. URL: https://www.mdpi.com/2313-433X/7/10/218?utm (дата звернення: 18.11.2025).
34. A Multi-Watermarking Algorithm for Medical Images Using Inception V3 and DCT. URL: https://www.sciencedirect.com/org/science/article/pii/S1546221822001710 (дата звернення: 19.11.2025).
35. Python як основа для систем тестування. URL: https://www.python.org/applications/testing (дата звернення: 19.11.2025).
36. Google Colab: the power of the cloud for machine learning. DataScientest. URL: https://datascientest.com/en/google-colab-the-power-of-the-cloud-for-machine-learning (дата звернення: 19.11.2025).
37. Google Colab Explained: Simplifying Your Workflow with Cloud Tools. Vast AI. URL: https://vast.ai/article/google-collab-explained-simplifying-your-workflow-with-cloud-tools (дата звернення: 19.11.2025).
38. Налаштування середовища виконання в Google Colab: поради та лайфхаки. FoxmindEd. URL: https://foxminded.ua/google-colab/ (дата звернення: 19.11.2025).
39. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.
40. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
Юрій ЯРЕМЧУК
"24" вересня 2025р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3

08-72.МКР.009.00.000.ТЗ

Керівник магістерської кваліфікаційної роботи
к.т.н., доц., доцент, завідувач кафедри МБІС
Карпінець В.В.

Вінниця - 2025 р.

1. **Найменування та область застосування.**

"Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3"

2. **Підстава для розробки**

Розробка виконується на основі наказу ректора № 313 від 24.09.2025 р.

3. **Мета та призначення**

3.1 Мета розробки: розробка алгоритму вбудовування та екстракції цифрового водяного знаку, що підвищує стійкість зображень у частотному просторі від частотних та геометричних атак за рахунок DCT та DWT перетворень, нейромережі Inception V3 та SIFT.

3.2 Призначення: забезпечення надійного захисту цифрових зображень від несанкційного використання та атак шляхом вдосконалення методу ЦВЗ.

4. **Джерела розробки**

4.1 Medical Image Hybrid Watermark Algorithm Based on Frequency Domain Processing and Inception v3. URL: https://advanced.onlinelibrary.wiley.com/doi/full/10.1002/aisy.202400654?utm_source=chatgpt.com (дата звернення: 09.11.2025);

4.2 A Hybrid Robust Image Watermarking Method Based on DWT-DCT and SIFT for Copyright Protection. URL: https://www.mdpi.com/2313-433X/7/10/218?utm (дата звернення: 09.11.2025).

5. **Вимоги до програми**

5.1 Вимоги до функціональних характеристик:

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

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

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

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

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять

– не менше 512 Mb;

– середовище функціонування

– операційна система сімейство Windows;

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

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розробленого програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

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

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

9. Стадії та етапи розробки

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

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

- ПЗ до магістерської кваліфікаційної роботи;
- програмний додаток;
- презентація; відзив керівника роботи;
- відзив опонента

Технічне завдання до виконання прийняв _____ Прокопович-Гузенко Л.В.

Додаток Б. Лістинг програми

```
# Встановіть OpenCV Contrib, якщо SIFT недоступний
import subprocess
import sys

try:
    import cv2
    cv2.SIFT_create()
except (ImportError, AttributeError):
    print("Installing opencv-contrib-python for SIFT...")
    subprocess.check_call([sys.executable, "-m", "pip", "install", "opencv-contrib-python"])
    print("\n--- IMPORTANT ---")
    print("Installation complete. Please restart the Colab runtime now and run the script again.")
    print("Go to 'Runtime' -> 'Restart runtime' in the menu.")
    sys.exit()

try:
    from skimage.metrics import structural_similarity as ssim_metric
except ImportError:
    print("Installing scikit-image for SSIM...")
    subprocess.check_call([sys.executable, "-m", "pip", "install", "scikit-image"])
    from skimage.metrics import structural_similarity as ssim_metric

import cv2
import numpy as np
import pywt
import tensorflow as tf
from tensorflow.keras.applications.inception_v3 import InceptionV3, preprocess_input
from tensorflow.keras.models import Model
from google.colab import files
from IPython.display import display, Image
from scipy.stats import mode

def upload_image(title="Choose an image"):
    """Відкриває інтерфейс завантаження файлів Google Colab і повертає завантажене зображення."""
    print(title)
    uploaded = files.upload()
```

```
    if not uploaded:
        raise ValueError("No image was uploaded.")
    filename = next(iter(uploaded))
    img = cv2.imdecode(np.frombuffer(uploaded[filename], np.uint8), cv2.IMREAD_UNCHANGED)
    if img is None:
        raise ValueError(f"Could not decode image: {filename}")
    return img

def logistic_map_sequence(seed, size, a=3.99):
    """Генерує хаотичну послідовність за допомогою логістичної карти."""
    x = np.zeros(size)
    x[0] = seed
    for i in range(1, size):
        x[i] = a * x[i-1] * (1 - x[i-1])
    return x

def encrypt_watermark(watermark, key_seed, threshold=0.5):
    """Шифрує водяний знак за допомогою хаотичної послідовності та операції XOR."""
    watermark_flat = watermark.flatten()
    chaotic_sequence = logistic_map_sequence(key_seed, len(watermark_flat))
    binary_chaotic = (chaotic_sequence > threshold).astype(np.uint8)
    encrypted_watermark_flat = np.bitwise_xor(watermark_flat, binary_chaotic)
    return encrypted_watermark_flat.reshape(watermark.shape)

def get_inception_importance_map(image, model):
    """Генерує карту важливості за допомогою InceptionV3."""
    img_resized = cv2.resize(image, (299, 299))
    # Переконайтеся, що зображення є 3-канальним для InceptionV3
    if len(img_resized.shape) == 2:
        img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_GRAY2BGR)
    else:
        img_rgb = img_resized
    img_preprocessed = preprocess_input(img_rgb)
    img_expanded = np.expand_dims(img_preprocessed, axis=0)
    feature_map = model.predict(img_expanded)
    importance_map = np.mean(feature_map[0], axis=-1)
        importance_map = (importance_map - importance_map.min()) / (importance_map.max() -
importance_map.min())
```

```
    return importance_map

def embed_watermark(host_image, watermark_image, model, key_seed=0.123, alpha_base=30, redundancy=3):
    """Вбудовує водяний знак за допомогою надійного 2-рівневого підходу DWT-DCT."""
    if len(host_image.shape) == 3:
        host_gray = cv2.cvtColor(host_image, cv2.COLOR_BGR2GRAY)
    else:
        host_gray = host_image
    if len(watermark_image.shape) == 3:
        watermark_gray = cv2.cvtColor(watermark_image, cv2.COLOR_BGR2GRAY)
    else:
        watermark_gray = watermark_image

    # Функції SIFT для геометричної корекції
    sift = cv2.SIFT_create()
    kp_host, des_host = sift.detectAndCompute(host_gray, None)

    # 2-рівневий DWT
    coeffs = pywt.wavedec2(host_gray, 'haar', level=2)
    cA2, (cH2, cV2, cD2), (cH1, cV1, cD1) = coeffs

    # Вбудувати в піддіапазон HL2 (cV2)
    band_to_embed = cV2

# Підготуйте водяний знак
    block_size = 8
    embeddable_blocks = (band_to_embed.shape[0] // block_size) * (band_to_embed.shape[1] // block_size)
        watermark_resized = cv2.resize(watermark_gray, (int(np.sqrt(embeddable_blocks / redundancy)), int(np.sqrt(embeddable_blocks / redundancy))))
    _, watermark_binary = cv2.threshold(watermark_resized, 127, 1, cv2.THRESH_BINARY)

    encrypted_wm = encrypt_watermark(watermark_binary, key_seed)

    # Резервування
    watermark_flat_redundant = np.repeat(encrypted_wm.flatten(), redundancy)

    if len(watermark_flat_redundant) > embeddable_blocks:
```

```
        raise ValueError(f"Watermark with redundancy ({len(watermark_flat_redundant)} bits) is too large for the host image ({embeddable_blocks} blocks).")

    # Карта важливості InceptionV3
    importance_map = get_inception_importance_map(host_gray, model)
                    importance_map_resized = cv2.resize(importance_map, (band_to_embed.shape[1], band_to_embed.shape[0]))

    # Адаптивне вбудовування
    wm_idx = 0
    band_watermarked = np.copy(band_to_embed)

    # Визначити пару коефіцієнтів DCT
    c1_pos, c2_pos = (2, 5), (3, 4)

    for i in range(0, band_to_embed.shape[0] - block_size + 1, block_size):
        for j in range(0, band_to_embed.shape[1] - block_size + 1, block_size):
            if wm_idx >= len(watermark_flat_redundant):
                break

            block = band_to_embed[i:i+block_size, j:j+block_size]
            dct_block = cv2.dct(block.astype(np.float32))

            local_importance = np.mean(importance_map_resized[i:i+block_size, j:j+block_size])
            alpha = alpha_base * (1 + local_importance)

            # Вбудовування відносин
            c1 = dct_block[c1_pos]
            c2 = dct_block[c2_pos]
            mean_val = (c1 + c2) / 2

            if watermark_flat_redundant[wm_idx] == 1:
                dct_block[c1_pos] = mean_val + alpha / 2
                dct_block[c2_pos] = mean_val - alpha / 2
            else: # bit is 0
                dct_block[c1_pos] = mean_val - alpha / 2
                dct_block[c2_pos] = mean_val + alpha / 2
```

```
            idct_block = cv2.idct(dct_block)
            band_watermarked[i:i+block_size, j:j+block_size] = idct_block
            wm_idx += 1
        if wm_idx >= len(watermark_flat_redundant):
            break

    # Інверсний 2-рівневий DWT
    coeffs_wm = cA2, (cH2, band_watermarked, cD2), (cH1, cV1, cD1)
    watermarked_image = pywt.waverec2(coeffs_wm, 'haar')
    watermarked_image = np.clip(watermarked_image, 0, 255).astype(np.uint8)

    return watermarked_image, encrypted_wm, (kp_host, des_host)

def correct_geometric_distortion(attacked_image, original_kp, original_des, original_shape):
    """Виправляє геометричні спотворення за допомогою зіставлення ознак SIFT."""
    sift = cv2.SIFT_create()

    if len(attacked_image.shape) == 3:
        attacked_gray = cv2.cvtColor(attacked_image, cv2.COLOR_BGR2GRAY)
    else:
        attacked_gray = attacked_image.astype(np.uint8)

    attacked_kp, attacked_des = sift.detectAndCompute(attacked_gray, None)

    if original_des is None or attacked_des is None or len(attacked_des) < 2:
        print("Попередження: недостатньо функцій SIFT. Зміна розміру як запасний варіант.")
        return cv2.resize(attacked_gray, (original_shape[1], original_shape[0]))

    FLANN_INDEX_KDTREE = 1
    index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
    search_params = dict(checks=50)
    flann = cv2.FlannBasedMatcher(index_params, search_params)
    matches = flann.knnMatch(original_des, attacked_des, k=2)

    good_matches = [m for m, n in matches if m.distance < 0.75 * n.distance]
MIN_MATCH_COUNT = 10
    if len(good_matches) >= MIN_MATCH_COUNT:
        src_pts = np.float32([original_kp[m.queryIdx].pt for m in good_matches]).reshape(-1, 1, 2)
```

```
        dst_pts = np.float32([attacked_kp[m.trainIdx].pt for m in good_matches]).reshape(-1, 1, 2)
        M, _ = cv2.findHomography(dst_pts, src_pts, cv2.RANSAC, 5.0)
        if M is not None:
            corrected_image = cv2.warpPerspective(attacked_gray, M, (original_shape[1], original_shape[0]))
            print("Geometric correction applied successfully.")
            return corrected_image

    print(f"Попередження: Не знайдено достатньо відповідних збігів або гомографія не вдалася. Зміна розміру як запасний варіант.")
    return cv2.resize(attacked_gray, (original_shape[1], original_shape[0]))

def extract_watermark_from_image(corrected_attacked_image, watermark_shape, redundancy=3):
    """Відновлює зашифрований водяний знак з геометрично скоригованого зображення.."""
    # 2-рівневий DWT на скоригованому зображенні
    coeffs_attacked = pywt.wavedec2(corrected_attacked_image, 'haar', level=2)
    _, (cH2_attacked, cV2_attacked, cD2_attacked), _ = coeffs_attacked
    band_attacked = cV2_attacked

    # Екстракція
    block_size = 8
    num_wm_bits = watermark_shape[0] * watermark_shape[1]
    extracted_bits_redundant = []

    c1_pos, c2_pos = (2, 5), (3, 4)

    for i in range(0, band_attacked.shape[0] - block_size + 1, block_size):
        for j in range(0, band_attacked.shape[1] - block_size + 1, block_size):
            if len(extracted_bits_redundant) >= num_wm_bits * redundancy:
                break

            block_attacked = band_attacked[i:i+block_size, j:j+block_size]
            dct_block_attacked = cv2.dct(block_attacked.astype(np.float32))

            diff = dct_block_attacked[c1_pos] - dct_block_attacked[c2_pos]
            extracted_bits_redundant.append(1 if diff > 0 else 0)

        if len(extracted_bits_redundant) >= num_wm_bits * redundancy:
            break
```

```
    # MAJORITY VOTING
    extracted_bits = []
    for i in range(0, len(extracted_bits_redundant), redundancy):
        chunk = extracted_bits_redundant[i:i+redundancy]
        if not chunk: continue
        bit, _ = mode(chunk)
        extracted_bits.append(bit)

    # Обробка випадків, коли вилучення дає менше бітів, ніж очікувалося
    if len(extracted_bits) < num_wm_bits:
        extracted_bits.extend([0] * (num_wm_bits - len(extracted_bits)))

    return np.array(extracted_bits).reshape(watermark_shape)

def extract_watermark(watermarked_image, original_host_image, watermark_shape, original_sift_features, key_seed=0.123, redundancy=3):
    """Витягує водяний знак і очищає зображення-носій, виконуючи геометричну корекцію тільки один раз."""
    # Виконайте геометричну корекцію один раз.
    kp_host, des_host = original_sift_features
    if len(original_host_image.shape) == 3:
        original_host_gray = cv2.cvtColor(original_host_image, cv2.COLOR_BGR2GRAY)
    else:
        original_host_gray = original_host_image
        corrected_watermarked_image = correct_geometric_distortion(watermarked_image, kp_host, des_host, original_host_gray.shape)

    # Частина 1: Розшифруйте витягнутий водяний знак
        encrypted_wm = extract_watermark_from_image(corrected_watermarked_image, watermark_shape, redundancy)
    watermark_flat = encrypted_wm.flatten()
    chaotic_sequence = logistic_map_sequence(key_seed, len(watermark_flat))
    binary_chaotic = (chaotic_sequence > 0.5).astype(np.uint8)
    decrypted_watermark_flat = np.bitwise_xor(watermark_flat, binary_chaotic)
    decrypted_watermark = decrypted_watermark_flat.reshape(watermark_shape)

    # Частина 2: Очистіть образ хоста, використовуючи той самий виправлений образ
```

```
    # Застосувати 2-рівневий DWT
    coeffs_wm = pywt.wavedec2(corrected_watermarked_image, 'haar', level=2)
    cA2, (cH2, cV2, cD2), (cH1, cV1, cD1) = coeffs_wm
    band_to_clean = cV2

    # Пройдіться по блоках, в які було вбудовано водяний знак, і нейтралізуйте сигнал.
    block_size = 8
    band_cleaned = np.copy(band_to_clean)
    c1_pos, c2_pos = (2, 5), (3, 4)

num_wm_bits = watermark_shape[0] * watermark_shape[1]
    blocks_to_process = (num_wm_bits * redundancy)
    block_idx = 0

    for i in range(0, band_to_clean.shape[0] - block_size + 1, block_size):
        for j in range(0, band_to_clean.shape[1] - block_size + 1, block_size):
            if block_idx >= blocks_to_process:
                break

            block = band_to_clean[i:i+block_size, j:j+block_size]
            dct_block = cv2.dct(block.astype(np.float32))

        # Процес вбудовування модифікує c1 і c2 на основі їх середнього значення.
            # Щоб змінити цю ситуацію, ми встановлюємо обидва коефіцієнти на їх середнє значення, ефективно
        # Видалення вбудованого сигналу різниці. Модель InceptionV3 та альфа.
            # Не потрібні для цього перетворення, оскільки середнє значення зберігається під час вбудовування.
            c1 = dct_block[c1_pos]
            c2 = dct_block[c2_pos]
            mean_val = (c1 + c2) / 2
            dct_block[c1_pos] = mean_val
            dct_block[c2_pos] = mean_val

            idct_block = cv2.idct(dct_block)
            band_cleaned[i:i+block_size, j:j+block_size] = idct_block
            block_idx += 1
        if block_idx >= blocks_to_process:
```

```
            break

    # Відновіть зображення за допомогою очищеної смуги DWT.
    coeffs_cleaned = cA2, (cH2, band_cleaned, cD2), (cH1, cV1, cD1)
    cleaned_image = pywt.waverec2(coeffs_cleaned, 'haar')
    cleaned_image = np.clip(cleaned_image, 0, 255).astype(np.uint8)

    return cleaned_image, decrypted_watermark * 255

def compute_psnr(original_img, watermarked_img):
    original_img = original_img.astype(np.float64)
    watermarked_img = watermarked_img.astype(np.float64)
    mse = np.mean((original_img - watermarked_img) ** 2)
    if mse == 0: return float('inf')
    return 20 * np.log10(255.0 / np.sqrt(mse))

def compute_BER(original_bits, recovered_bits):
    original_bits = original_bits.flatten()
    recovered_bits = recovered_bits.flatten()
    if len(original_bits) != len(recovered_bits):
        min_len = min(len(original_bits), len(recovered_bits))
        original_bits = original_bits[:min_len]
        recovered_bits = recovered_bits[:min_len]
        print(f"Warning: Bitstreams have different lengths. Comparing first {min_len} bits.")
    if len(original_bits) == 0: return 100.0
    error_bits = np.sum(original_bits != recovered_bits)
    return (error_bits / len(original_bits)) * 100

def compute_NCC(original, recovered):
    original = original.flatten().astype(np.float64)
    recovered = recovered.flatten().astype(np.float64)
    if len(original) == 0 or len(recovered) == 0 or len(original) != len(recovered): return 0
    original_mean, recovered_mean = np.mean(original), np.mean(recovered)
    numerator = np.sum((original - original_mean) * (recovered - recovered_mean))
    denominator = np.sqrt(np.sum((original - original_mean)**2) * np.sum((recovered - recovered_mean)**2))
    return numerator / denominator if denominator != 0 else 0

def compute_SSIM(original, recovered):
```

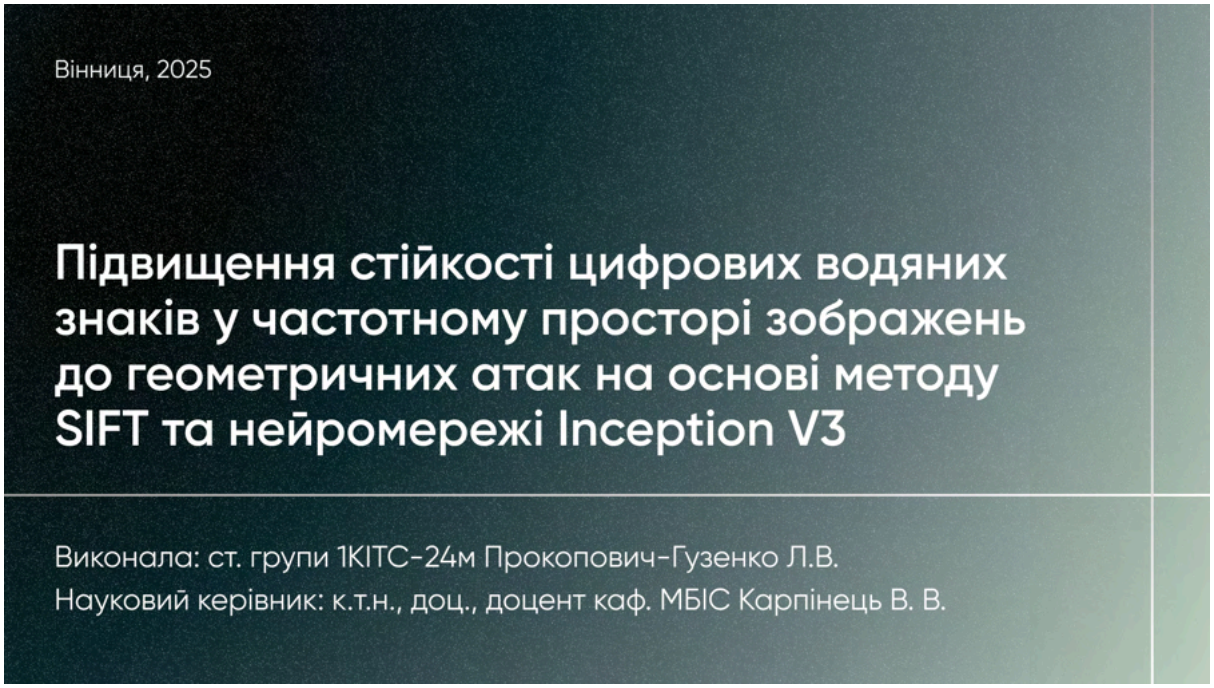
```
    return ssim_metric(original, recovered, data_range=recovered.max() - recovered.min())

def attack_jpeg(image, quality=40):
    _, encimg = cv2.imencode('.jpg', image, [int(cv2.IMWRITE_JPEG_QUALITY), quality])
    return cv2.imdecode(encimg, 0)

def attack_rotate(image, angle=15):
    (h, w) = image.shape[:2]
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, angle, 1.0)
    return cv2.warpAffine(image, M, (w, h))

def attack_resize(image, scale=0.7):
    (h, w) = image.shape[:2]
    resized = cv2.resize(image, (int(w * scale), int(h * scale)), interpolation=cv2.INTER_AREA)
    return cv2.resize(resized, (w, h), interpolation=cv2.INTER_CUBIC)

def attack_gaussian_noise(image, mean=0, var=100): # Increased variance
    sigma = var**0.5
    gaussian = np.random.normal(mean, sigma, image.shape)
    noisy_image = np.clip(image.astype(np.float32) + gaussian, 0, 255).astype(np.uint8)
    return noisy_image

def attack_salt_and_pepper(image, amount=0.02):
    output = np.copy(image)
    # Сіль
    num_salt = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_salt)) for i in image.shape]
    output[tuple(coords)] = 255
    # Перець
    num_pepper = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in image.shape]
    output[tuple(coords)] = 0
    return output

def  apply_attacks_and_evaluate(watermarked_image, original_host_image, encrypted_wm, model, key_seed,
original_sift_features):
    attacks = {
```

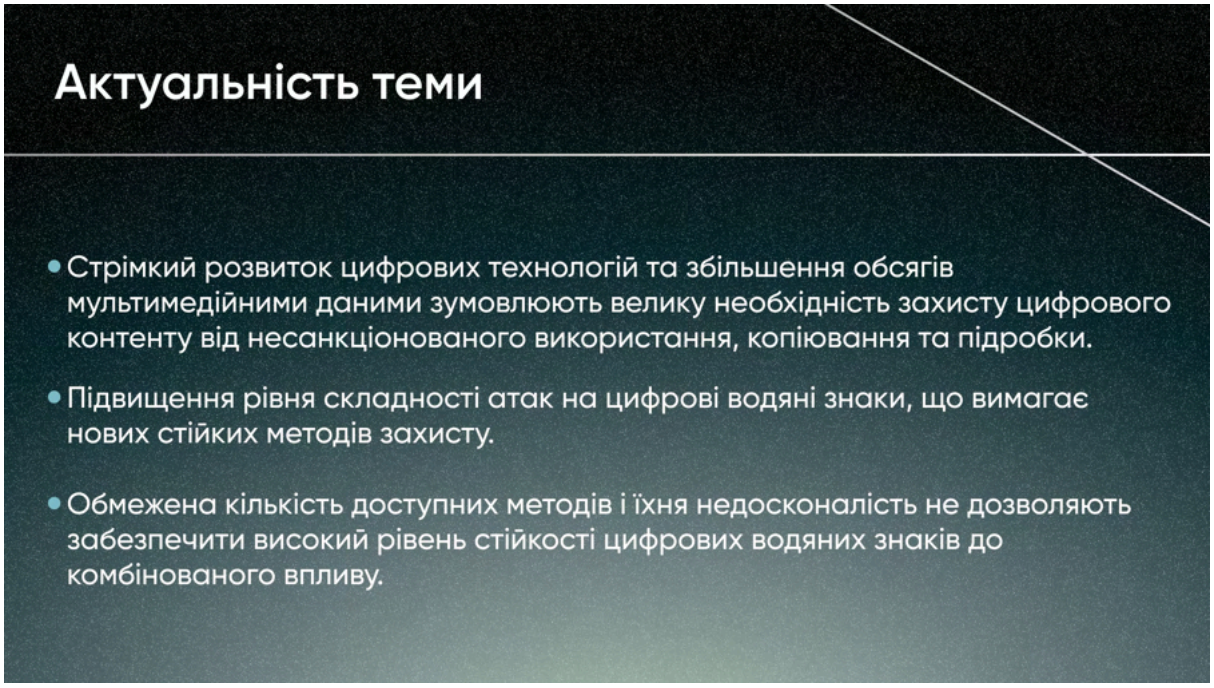
```
        "JPEG Compression (Q=40)": lambda img: attack_jpeg(img, quality=40),
        "Rotation (15 deg)": lambda img: attack_rotate(img, angle=15),
        "Resizing (70%)": lambda img: attack_resize(img, scale=0.7),
        "Gaussian Noise (Var=100)": attack_gaussian_noise,
        "Salt-and-Pepper (2%)": attack_salt_and_pepper
    }
    results = {}
    kp_host, des_host = original_sift_features
    if len(original_host_image.shape) == 3:
        original_host_gray = cv2.cvtColor(original_host_image, cv2.COLOR_BGR2GRAY)
    else:
        original_host_gray = original_host_image

    for attack_name, attack_func in attacks.items():
        print(f"\n--- Applying {attack_name} Attack ---")
        attacked_image = attack_func(watermarked_image)
        display_cv_image(attacked_image, f"Image after {attack_name}")

        # Виконайте геометричну корекцію перед вилученням
            corrected_attacked_image = correct_geometric_distortion(attacked_image, kp_host, des_host,
original_host_gray.shape)
                recovered_wm_encrypted = extract_watermark_from_image(corrected_attacked_image,
encrypted_wm.shape)

        display_cv_image(recovered_wm_encrypted.astype(np.uint8)*255, "Recovered (Encrypted) Watermark")

        ber = compute_BER(encrypted_wm, recovered_wm_encrypted)
        ncc = compute_NCC(encrypted_wm, recovered_wm_encrypted)
                            ssim_val = compute_SSIM(encrypted_wm.astype(np.uint8) * 255,
recovered_wm_encrypted.astype(np.uint8) * 255)

        results[attack_name] = {"BER": ber, "NCC": ncc, "SSIM": ssim_val}
        print(f"BER: {ber:.2f} %")
        print(f"NCC: {ncc:.4f}")
        print(f"SSIM: {ssim_val:.4f}")
    return results

def display_cv_image(image, title=""):
```

```
    """Функція для відображення зображень у Colab."""
    _, encoded_image = cv2.imencode('.png', image)
    display(Image(data=encoded_image.tobytes(), format='png'))
    print(title)

def main():
    """Основна функція для запуску процесу нанесення водяних знаків."""
    try:
        host_image = upload_image("Upload the HOST image (e.g., a photograph)")
        watermark_image = upload_image("Upload the WATERMARK image (e.g., a QR code or logo)")

        print("\nLoading InceptionV3 model...")
        base_model = InceptionV3(weights='imagenet', include_top=False)
        model = Model(inputs=base_model.input, outputs=base_model.get_layer('mixed7').output)
        print("Model loaded.")

        print("\nEmbedding watermark...")
                watermarked_image, encrypted_wm, original_sift_features = embed_watermark(host_image,
watermark_image, model)
        print("Embedding complete.")

            host_gray = cv2.cvtColor(host_image, cv2.COLOR_BGR2GRAY) if len(host_image.shape) == 3 else
host_image
        psnr_value = compute_psnr(host_gray, watermarked_image)

        print("\n--- Results ---")
        display_cv_image(host_image, "Оригінальне зображення хоста")
        display_cv_image(watermark_image, "Водяний знак зображення")
        display_cv_image(watermarked_image, "Зображення з водяним знаком")
        print(f"\nPSNR Value: {psnr_value:.2f} dB (Goal: >= 40 dB)")

        print("\n--- Видалення водяного знака з зображення з водяним знаком ---")
        cleaned_image, extracted_watermark = extract_watermark(
            watermarked_image,
            host_image.copy(),
            encrypted_wm.shape,
            original_sift_features
        )
```

```
        display_cv_image(cleaned_image, "Очищений образ хоста")
        display_cv_image(extracted_watermark, "Витягнутий водяний знак")
print("\n--- Оцінка стійкості---")
        key_seed = 0.123
        results = apply_attacks_and_evaluate(
            watermarked_image,
            host_image.copy(),
            encrypted_wm,
            model,
            key_seed,
            original_sift_features
        )

    except (ValueError, Exception) as e:
        import traceback
        print(f"An error occurred: {e}")
        traceback.print_exc()

if name == "main":
    main()
```

Додаток В. Ілюстративний матеріал





Мета роботи

Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак шляхом поєднання методу SIFT та нейромережі Inception V3 у процесі вбудовування і екстракції водяного знаку. Стрімкий розвиток цифрових технологій та збільшення обсягів мультимедійними даними зумовлюють велику необхідність захисту цифрового контенту від несанкціонованого використання, копіювання та підробки.

Елементи дослідження

Дискретне косинусне перетворення

Дискретне вейвлет перетворення

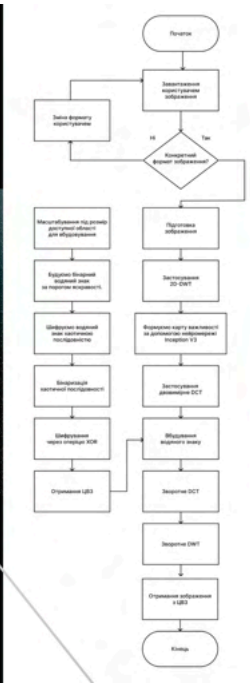
SIFT

Inception v3

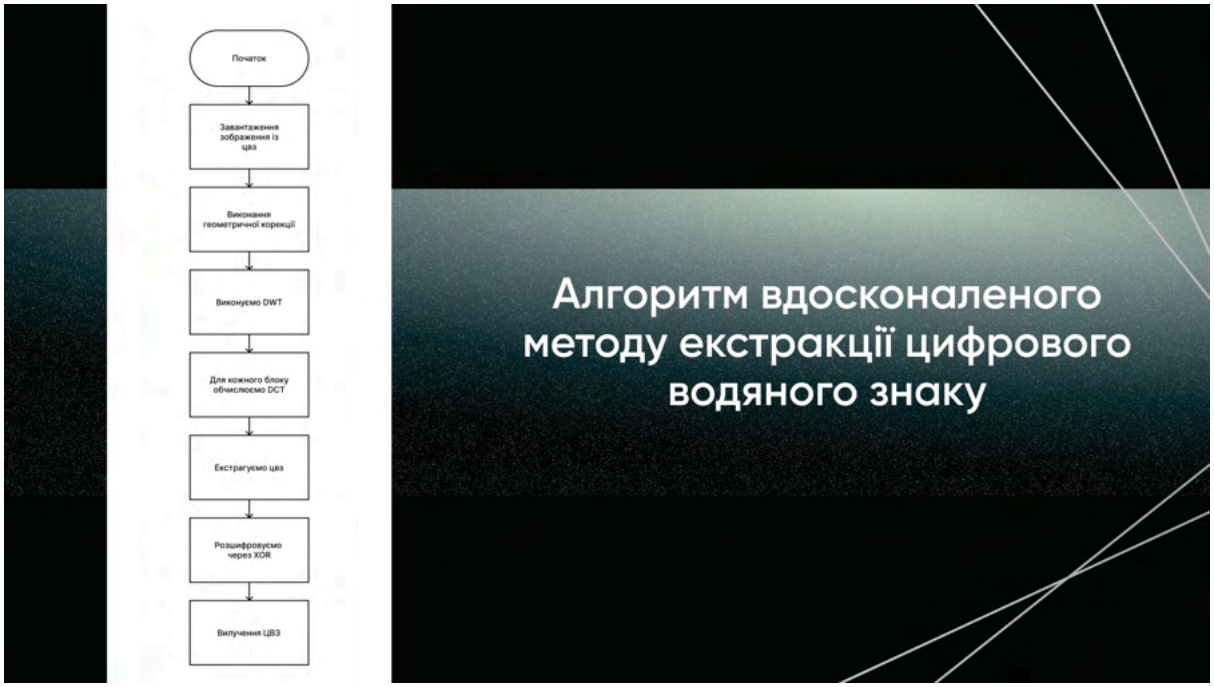
Дискретне косинусне перетворення (DCT)	Це математичне перетворення, яке представляє зображення не у вигляді пікселів, а у вигляді частотних складових, тобто показує, які частоти (деталі, текстури, контури) присутні в зображенні та з якою інтенсивністю.
Дискретне вейвлет перетворення (DWT)	Це математичний метод, який розкладає зображення на різні частоти та масштаби, дозволяючи бачити як загальну структуру, так і дрібні деталі одночасно. Простіше кажучи, DWT — це спосіб подивитися на зображення і крупним планом, і здалеку за допомогою хвилеподібних функцій — вейвлетів.
SIFT	Це метод, який знаходить особливі точки, що не змінюються при: повороті, масштабуванні, зміщенні, зміні яскравості, частковому обрізанні
Inception V3	Це глибока згорткова нейронна мережа (CNN), розроблена Google. Використовується для класифікації зображень та створення карти важливості.

Новизна роботи

У роботі запропоновано гібридний алгоритм цифрового водяного знакування, який поєднує частотні перетворення (DWT та DCT) з інваріантними ознаками SIFT та семантичною обробкою зображення за допомогою нейронної мережі Inception V3. Новизна полягає в тому, що водяний знак вбудовується адаптивно, з урахуванням змісту зображення та його найбільш захищених областей, а процес відновлення забезпечується за рахунок інваріантності до геометричних спотворень.

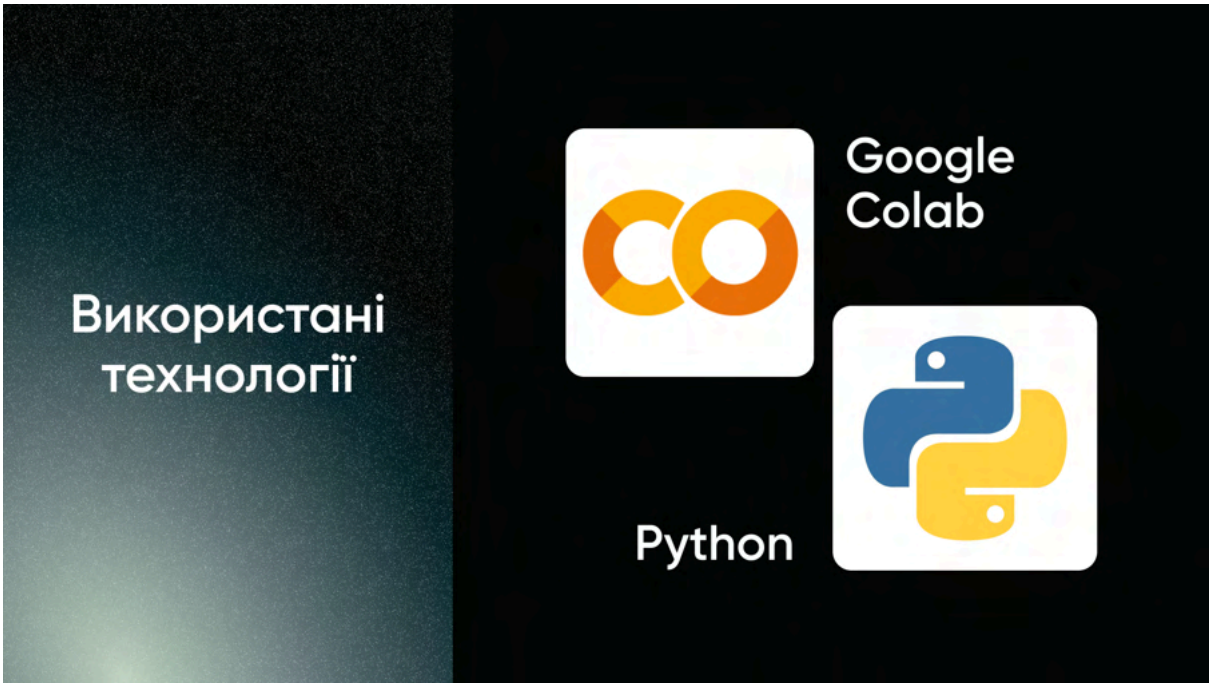


Алгоритм вдосконаленого методу вбудування цифрового водяного знаку


Початок
Завантаження зображення із цвз
Виконання геометричної корекції
Виконуємо DWT
Для кожного блоку обчислюємо DCT
Екстрагуємо цвз
Розшифровуємо через XOR
Вилучення ЦВЗ
Алгоритм вдосконаленого методу екстракції цифрового водяного знаку


Чому саме SIFT?
Метод SIFT використовується в роботі завдяки здатності виявляти та описувати ключові точки зображення, які залишаються стабільними навіть після значних геометричних спотворень. Його головною перевагою є інваріантність до масштабу, повороту, зсуву та часткової зміни освітлення, що робить його одним із найнадійніших алгоритмів у задачах, де структура зображення може бути порушена. SIFT дозволяє зберегти прив'язку до характерних точок зображення незалежно від того, як змінено його геометричну форму.


Чому саме Inception V3?
Використання нейромережі Inception V3 у розробленому алгоритмі зумовлене її здатністю глибоко аналізувати структуру зображення та визначати найбільш інформативні й стійкі зони для вбудовування цифрового водяного знака. Архітектура Inception V3 побудована на многомасштабних згорткових блоках, які вміють одночасно розпізнавати різні типи текстур, контурів та локальних структур. Завдяки цьому мережа може формувати карту важливості (importance map), що відображає, у яких ділянках зображення наявні високостійкі компоненти, здатні зберегти водяний знак без втрати якості зображення.


Використані технології
Google Colab
Python


Зображення
з водяним знаком
SIFT
ознаки
Зображення
з екстрагованим ВЗ
Екстрагований
водяний знак
Вихідні
зображення
для ЦВЗ


Вхідні зображення для ЦВЗ
Зображення
“Пейзаж”
Водяний знак
“Символ”

Таблиця метрик, дія атак на зображення з водяним знаком

Атака	BER (%)	NCC	SSIM
JPEG-стиснення (Q=40)	0.00	1.0000	1.0000
Обертання 15°	3.40	0.9343	0.9635
Масштабування 70%	0.00	1.0000	1.0000
Гаусовий шум (Var=100)	0.00	1.0000	1.0000
“Сольовий” шум 2%	4.94	0.9018	0.9123

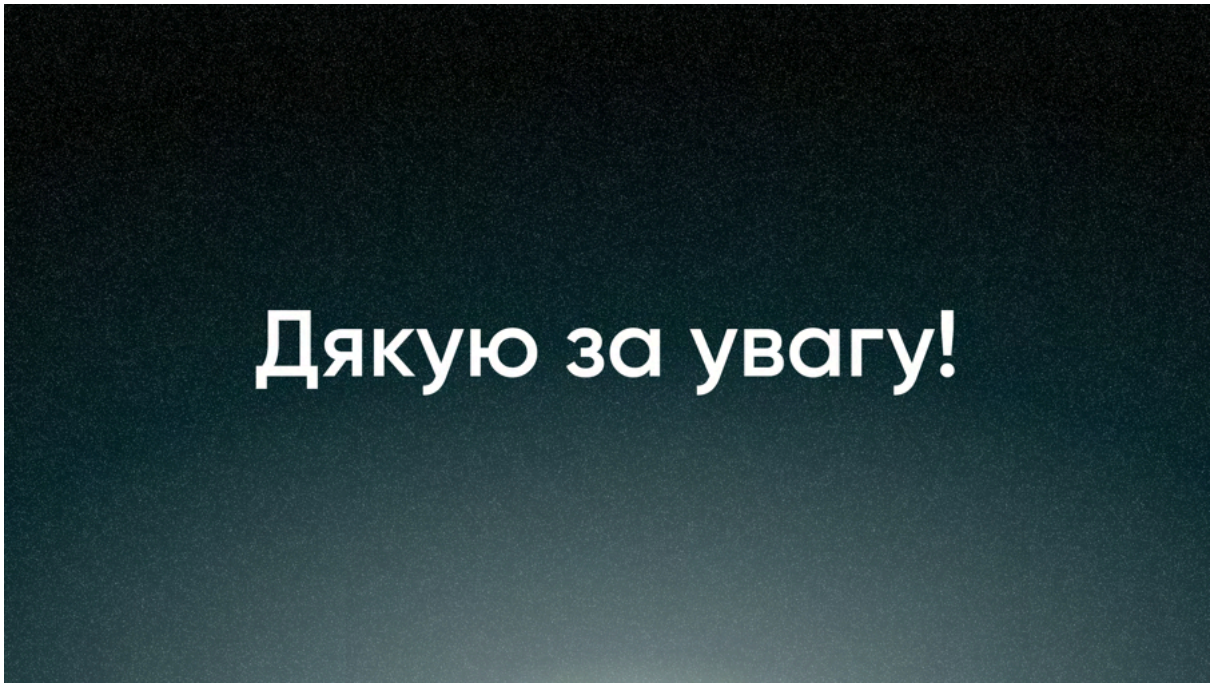
Удосконалений метод

Атака	BER (%)	NCC	SSIM
JPEG-стиснення (Q=40)	1.87	0.9621	0.9478
Обертання 15°	12.45	0.8154	0.8642
Масштабування 70%	6.57	0.8954	0.9183
Гаусовий шум (Var=100)	2.78	0.9530	0.9387
“Сольовий” шум 2%	14.63	0.7420	0.8125

Аналог-метод

Висновки

У даній дипломній роботі були досліджені основи цифрового водяного знаку, розроблено та реалізовано вдосконалений метод підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3


Дякую за увагу!

Додаток Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Підвищення стійкості цифрових водяних знаків у частотному просторі зображень до геометричних атак на основі методу SIFT та нейромережі Inception V3

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем факультет менеджменту та інформаційної безпеки
гр.1КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **2,17 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту**
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

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

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

Особа, відповідальна за перевірку Коваль Н.П. _____

З висновком експертної комісії ознайомлений(-на)

Керівник _____ доц. Карпінець В.В.

Здобувач _____ Прокопович-Гузенко Л.В.