



Г. Ю. Терещенко<sup>1</sup>, І. В. Кириченко<sup>2</sup>, К. С. Смеляков<sup>3</sup>, А. Є. Олійник<sup>4</sup>

<sup>1</sup>ХНУРЕ, м. Харків, Україна, hlib.tereshchenko@nure.ua, ORCID iD: 0000-0001-8731-2135

<sup>2</sup>ХНУРЕ, м. Харків, Україна, iryna.kyrychenko@nure.ua, ORCID iD: 0000-0002-7686-6439

<sup>3</sup>ХНУРЕ, м. Харків, Україна, kyrylo.smelyakov@nure.ua, ORCID iD: 0000-0001-9938-5489

<sup>4</sup>ХНУРЕ, м. Харків, Україна, artem.oliinyk@nure.ua, ORCID iD: 0009-0008-8613-964X

## АНАЛІЗ МЕТОДІВ СТИСНЕННЯ ЗОБРАЖЕНЬ ДЛЯ ЗБЕРІГАННЯ У ДЕЦЕНТРАЛІЗОВАНИХ БЛОКЧЕЙН СХОВИЩАХ

Розглянуто методи стиснення зображень для зберігання у децентралізованих блокчейн-сховищах, зокрема в системі IPFS. Пропонується методологія оцінки ефективності різних алгоритмів стиснення зображень, що включає як стиснення без втрат (PNG, TIFF, GIF), так і стиснення з втратами (JPEG, WEBP). Виконується порівняння показників розміру файлів, якості зображень (за допомогою метрики PSNR), часу завантаження в IPFS та пропускну здатності для кожного методу стиснення. Для кожного зображення вимірюється хеш, який зберігається у смарт-контракті в локальній блокчейн-мережі, розгорнутій за допомогою Ganache. Використання таких методів дозволяє оптимізувати зберігання та передачу зображень у блокчейн-сховища, забезпечуючи баланс між якістю зображень, розміром файлів та швидкістю доступу.

БЛОКЧЕЙН, СТИСНЕННЯ ЗОБРАЖЕНЬ, СХОВИЩА ДАНИХ, IPFS, GANACHE, ГРАФІКА.

**G.Yu. Tereshchenko, I.V. Kyrychenko, K.S. Smelyakov, A.E. Oliynyk. Analysis of image compression methods for storage in decentralized blockchain repositories.** The methods of image compression for storage in decentralized blockchain repositories, specifically in the IPFS system, are examined. A methodology for evaluating the efficiency of different image compression algorithms is proposed, incorporating both lossless compression (PNG, TIFF, GIF) and lossy compression (JPEG, WEBP). The comparison includes metrics such as file size, image quality (measured by PSNR), upload time to IPFS, and bandwidth for each compression method. Each image is hashed, and the hash is stored in a smart contract within a local blockchain network deployed using Ganache. Such methods enable the optimization of image storage and transfer within blockchain repositories, ensuring a balance between image quality, file size, and access speed.

BLOCKCHAIN, IMAGE COMPRESSION, DATA STORAGE, IPFS, GANACHE, GRAPHICS.

### Вступ

Сучасний науковий прогрес активно просуває технології, випереджаючи їхній розвиток і впровадження нововведень. Раніше для зберігання великого обсягу зображень доводилося витратити значний час і ресурси. Тепер майже кожна компанія спрямована на поліпшення процесу зберігання шляхом інтеграції систем стиснення зображень у свої продукти.

Системи стиснення здатні працювати з різними типами даних, такими як зображення, звук, відео, текст тощо. Вони відрізняються одна від одної своєю архітектурою, типом пам'яті, тощо. Актуальність даної теми полягає в тому, що за останні роки в галузі зберігання даних зростає потреба у надійних і безпечних методах забезпечення конфіденційності та цілісності інформації. В контексті розвитку технологій зображень, виникає велика потреба у збереженні та обміні цими даними в безпечних та надійних умовах. Таким чином, вивчення методів стиснення зображень для їх зберігання у блокчейн-сховищі має велике практичне значення у сучасному інформаційному середовищі. У цьому дослідженні вирішено дослідити методи стиснення зображень для їх ефективного зберігання у блокчейн-сховищі оскільки технологія блокчейну [1] пропонує ряд переваг для зберігання

даних, таких як підвищена безпека, надійність та масштабованість.

На сьогоднішній день існує багато доступних систем стиснення зображень, які відрізняються за якістю в залежності від алгоритму. Якість стиснення прямо пропорційна якості алгоритму, який, у свою чергу, залежить від розміру, типу та обсягу даних, а також від типу системи, в якій він застосовується.

Метою роботи є визначення найбільш ефективного методу стиснення зображень для зберігання в блокчейн сховищах з точки зору розміру файлу, якості зображення та продуктивності передачі даних.

Об'єктом дослідження є ефективність методів стиснення зображень, які використовуються для зберігання у блокчейн сховищах. Предметом дослідження є процес стиснення зображень та його вплив на розмір файлів та якість зображення.

Методами дослідження є проведення експериментів зі стисненням зображень різними методами та аналіз результатів за допомогою спеціалізованих програмних інструментів.

Отримані результати можуть бути використані для покращення процесу зберігання зображень у блокчейн сховищах, а також можуть послужити основою для подальших досліджень у галузі стиснення даних у децентралізованих системах.

**1. Технології збереження зображень у блокчейн сховищах**

Зображення, які використовуються в сучасних інформаційних системах, зазвичай мають великий обсяг даних і потребують ефективних методів зберігання та передачі. У зв'язку з цим, блокчейн сховища активно досліджуються для зберігання зображень, оскільки вони пропонують децентралізовану, безпечну та прозору інфраструктуру для зберігання даних [2]. В даному дослідженні розглядається найпопулярніше блокчейн сховище — IPFS.

IPFS активно використовується як протокол комунікації та розподілу даних, побудований на основі блокчейн технології. Це блокчейн сховище, що пропонує децентралізовану архітектуру для зберігання та обміну даними [3].

При виборі IPFS для зберігання зображень у блокчейн сховищах враховуються кілька важливих чинників. IPFS вирізняється своєю децентралізованістю та масштабованістю, що робить його привабливим варіантом для зберігання даних. Ця характеристика широко поєднується з основними принципами блокчейн технології, забезпечуючи надійність та безпеку даних.

Архітектура IPFS базується на концепції розподіленої мережі з додатковим шаром блокчейн технології. Кожен вузол мережі може бути як клієнтом, що зберігає дані, так і постачальником контенту [4]. Коли клієнт запитує доступ до певного файлу за його хешем, мережа автоматично знаходить найближчий вузол, який має цей файл, та передає його клієнту через блокчейн (див. рис. 1).

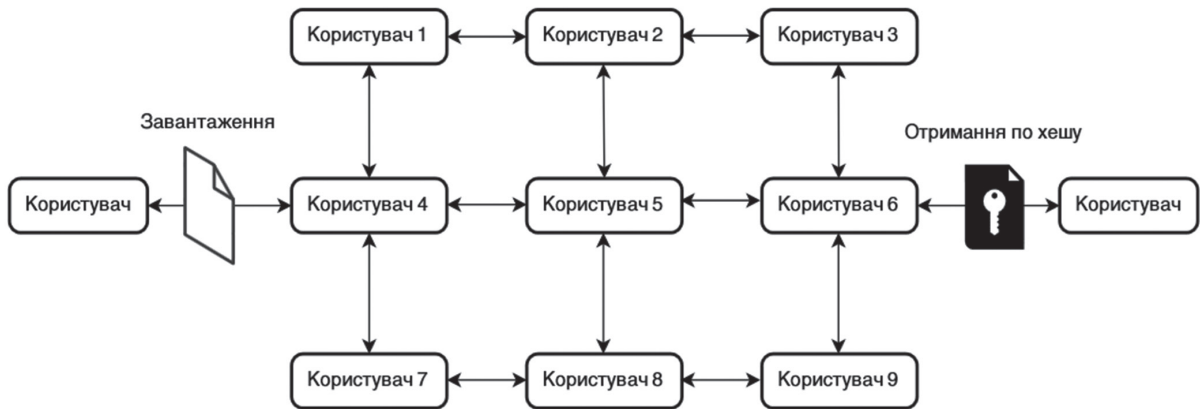


Рис. 1. Архітектура сучасних блокчейн сховищ

Додатково, в сучасному світі IPFS активно використовують в інтеграції з блокчейн технологією, що дозволяє забезпечити прозорість та цілісність збережених даних. Оскільки кожен файл отримує

унікальний хеш [5], він може бути легко записаний в блокчейн, забезпечуючи незмінність даних та перевірку їхньої цілісності (див. рис. 2).



Рис. 2. Взаємодія IPFS з блокчейн мережею

Враховуючи постійний розвиток IPFS та активні дослідження в галузі блокчейн технологій, обрання цього блокчейн сховища для зберігання зображень відкриває широкі перспективи для подальшого розвитку та вдосконалення цієї інфраструктури.

Незважаючи на переваги, використання IPFS як блокчейн сховища також має свої виклики, такі як обмежена пропускна здатність мережі. Це обов'язково враховується при виборі методів стиснення (див. табл.1).

Таблиця 1

Актуальні переваги та недоліки зберігання зображень в децентралізованих сховищах

Характеристика	Плюси	Мінуси
Децентралізованість	Забезпечує високий рівень безпеки та надійності даних, оскільки дані розподіляються по різних вузлах мережі	Обмежена пропускна здатність, особливо в глобальній мережі
Масштабованість	Здатний ефективно обробляти великі обсяги даних та взаємодіяти з великою кількістю користувачів	Проблеми з пропускною здатністю при великих обсягах даних
Унікальність хешу	Кожен файл отримує унікальний хеш, що гарантує цілісність та безпеку даних	Можливість втрати доступу до даних у разі втрати хешу
Прозорість даних	Забезпечує прозорість та відкритий доступ до даних	Проблеми з конфіденційністю даних та приватністю
Вартість зберігання	Вартість зберігання даних зазвичай нижча порівняно з централізованими рішеннями	Витрати на мережеву активність та маніпуляції даними

Отже, вибір відповідних методів стиснення є ключовим для забезпечення ефективного, економічного та безпечного зберігання зображень у блокчейн сховищах.

**2. Технології стиснення зображень**

Технології стиснення зображень можна розділити на два стиснення з втратами та стиснення без втрат.

Стиснення з втратати - метод, який дозволяє значно зменшити розмір файлу за рахунок втрати якості зображення. Цей метод дозволяє досягти значного зменшення розміру файлу без помітних втрат якості;

Стиснення без втрат - метод, який дозволяє зберегти початкову якість зображення. Однак, цей метод, як правило, не дозволяє досягти такого значного зменшення розміру файлу, як стиснення з втратами.

До плюсів втратного стиснення можна віднести високу ефективність. Втратне стиснення може забезпечити значно більший ступінь стиснення, ніж безвтратне. Це може призвести до значної економії місця на диску, підвищення пропускної здатності мережі та покращення ефективності обробки зображень.

Тим часом до мінусів втратного стиснення зазвичай відносять втрату якості. Втратне стиснення передбачає видалення деякої інформації з зображення. Це може призвести до втрати деталей, зміни кольору

або контрасту. Також серед мінусів розглядають неможливість відновлення даних. Втративши інформацію, яку було видалено при стисканні, її неможливо відновити.

Втратне стиснення широко застосовується для обробки зображень, де незначна втрата деталізації є допустимою і не впливає суттєво на їх сприйняття. Використання методів втратного стиснення, дозволяє значно зменшити розмір файлу за рахунок часткового зменшення якості зображення, що є прийнятним компромісом у багатьох випадках. Ці методи базуються на алгоритмах, що відкидають або узагальнюють деякі деталі зображення, які менш помітні для людського ока, зберігаючи при цьому загальну візуальну якість.

Тим часом до плюсів безвтратного стиснення відносять збереження якості. Безвтратне стиснення не призводить до втрати інформації з зображення. Це означає, що стиснене зображення буде повністю відповідати оригінальному.

Але основний мінус безвтратного стиснення це низька ефективність. Безвтратне стиснення зазвичай забезпечує нижчий ступінь стиснення, ніж втратне. Це означає, що для зберігання стиснених зображень може знадобитися більше місця на диску або пропускної здатності мережі.

Безвтратне стиснення використовується в різних областях, де збереження якості зображень є вирішальним аспектом. Наприклад, у медичній сфері або в обробці зображень для друку, важливо зберегти всі важливі деталі та точність, тому безвтратні методи стиснення є невід'ємною частиною процесу стиснення зображень. Використання безвтратного стиснення дозволяє зменшити обсяг даних, зберігаючи при цьому високий рівень якості, що робить його незамінним інструментом у сучасній обробці зображень.

Наразі вибір методу стиснення залежить від конкретного застосування. Для зображень з високою деталізацією, для яких втрата якості не є допустимою, слід використовувати безвтратне стиснення. Для зображень, для яких важлива пропускна здатність або розмір файлу, можна використовувати втратне стиснення

Після того, як ми зрозуміли актуальність дослідження методів стиснення зображень у контексті їх зберігання в блокчейн-сховищах, наступним кроком є аналіз доступних алгоритмів стиснення. Розглянемо основні методи, які широко використовуються для зменшення розміру зображень без втрати якості та з втратами, щоб визначити їх ефективність у контексті наших досліджень.

У дослідженні розглядаються алгоритми втратного стиснення, такі як JPG, WEBP та алгоритми стиснення без втрат, такі як PNG, TIFF, GIF.

### 3. Метод стиснення PNG

Метод стиснення PNG є форматом зображень, який активно використовується для зберігання растрових зображень [6]. Однією з ключових особливостей PNG є можливість стиснення без втрат якості. Для цього реалізований алгоритм стиснення Deflate.

Deflate – це алгоритм стиснення з втратами, що використовується в форматі PNG для зменшення розміру файлу без втрати якості. Він базується на комбінації алгоритму LZ77 та кодування Хаффмана [7].

Алгоритм LZ77 використовується для заміни повторюваних послідовностей байтів на зворотні посилання на знайдені раніше послідовності в тексті. Цей процес ґрунтується на вікні ковзання, яке зберігає копії останніх даних. Замість зберігання повної копії, використовується посилання на позицію та довжину послідовності.

Після застосування LZ77 до даних, отримані символні послідовності кодуються за допомогою кодування Хаффмана, яке стискає часто повторювані символи у менші біти та рідко повторювані символи у більші.

Початковий етап полягає у визначенні ймовірностей появи кожного символу у зображенні. У контексті стиснення зображень ці символи можуть бути представлені, наприклад, значеннями пікселів або іншими кодами, що представляють кольори.

Для обчислення ймовірностей використовується формула 1:

$$p_i = \frac{n_i}{N} \quad (1)$$

де  $p_i$  – ймовірність появи символу  $i$ ,  $n_i$  – кількість входжень символу  $i$  у зображенні,  $N$  – загальна кількість символів у зображенні.

Після обчислення ймовірностей будується дерево Хаффмана. Для кожного символу будується листок дерева з вагою, що відповідає ймовірності появи символу. Внутрішні вузли мають вагу, яка є сумою ваг дітей. Після побудови дерева вузлам надаються ваги, відповідні їх ймовірностям.

Для призначення кодів Хаффмана кожному символу призначається унікальний код, що визначається шляхом від кореня дерева до відповідного листка. Код складається з послідовності бітів, де 0 вказує на ліве відгалуження, а 1 – на праве.

Після цього зображення кодується, замінюючи кожен символ його відповідним кодом. Це призводить до створення бітової послідовності, яка представляє стиснене зображення.

Наприклад для побудови дерева Хаффмана для даних «AAAABBBBBBCCCCDDDD» після аналізу частоти, буде отримано наступні значення:

- A: частота 4 рази, ймовірність появи 4/16;

- B: частота 5 разів, ймовірність появи 5/16;
- C: частота 4 рази, ймовірність появи 4/16;
- D: частота 3 рази, ймовірність появи 3/16.

Після цього визначається вага кожного листка і будується дерево Хаффмана (див. рис. 3).

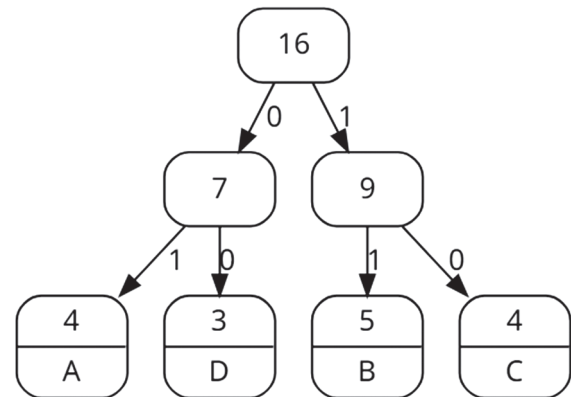


Рис. 3. Приклад діаграми Хаффмана

Таким чином, метод стиснення PNG забезпечує зменшення розміру файлу без втрати якості. Це дозволяє ефективно зберігати та передавати зображення, зберігаючи всі деталі та точність. В сучасному світі використання цього методу є критичним для сфер медичної діагностики, наукових досліджень, поліграфії та інших сфер де важлива висока якість зображення.

### 4. Метод стиснення TIFF

Метод стиснення TIFF є форматом зображень, який використовується для зберігання растрових зображень. Однією з ключових особливостей TIFF є можливість стиснення без втрат якості. Для цього використовується алгоритм стиснення LZW.

Алгоритм LZW – це алгоритм стиснення без втрат, що використовується в форматі TIFF для зменшення розміру файлу без втрати якості. Він базується на використанні словника фіксованого розміру, який динамічно оновлюється під час процесу стиснення [8].

Першим етапом є створення словника, який містить всі можливі символи вхідних даних. Під час процесу стиснення, алгоритм шукає найдовшу послідовність символів, яка вже є в словнику, і замінює її на відповідний код словника. Ця послідовність символів потім розширюється на один символ, і нова послідовність додається до словника.

Наприклад, розглянемо стиснення рядка «АВАВАВА» за допомогою LZW. Спочатку словник містить лише окремі символи «А» з кодом «1» та «В» з кодом «2».

Під час стиснення, алгоритм спочатку знаходить «А» в словнику і виводить відповідний код. Потім він розширює послідовність до «АВ», якої ще немає в словнику, тому «АВ» додається до словника з кодом «3», а код для «А» виводиться.

Процес продовжується, поки весь рядок не буде стиснутий, і в результаті отримується послідовність кодів словника. Таким чином, вихідний код для рядка «АВАВАВА» за допомогою алгоритму LZW буде 1, 2, 3, 1, 3 (див. табл. 1.2).

**Таблиця 2**  
**Покроковий принцип роботи алгоритму LZW**

Крок	Словник (символ : код)	Вивід алгоритму
1	A : 1, B : 2, AB : 3	1
2	A : 1, B : 2, AB : 3, BA : 4	1, 2
3	A : 1, B : 2, AB : 3, BA : 4, ABA : 5	1, 2, 3
4	A : 1, B : 2, AB : 3, BA : 4, ABA : 5	1, 2, 3, 1
5	A : 1, B : 2, AB : 3, BA : 4, ABA : 5	1, 2, 3, 1, 3

Цей алгоритм широко використовується для стиснення даних, які містять повторювані послідовності символів, і дозволяє стискати зображення без втрат. Важливо зазначити, що, хоча LZW є алгоритмом стиснення без втрат, він може бути неефективним для стиснення деяких типів даних. Наприклад, він може не зменшувати розмір файлу, якщо дані містять високий рівень шуму або якщо повторювані послідовності символів є дуже короткими або рідкісними.

### 5. Метод стиснення GIF

Метод стиснення GIF є форматом зображень, який використовується для зберігання растрових зображень. Однією з ключових особливостей GIF є можливість стиснення без втрат якості. Для цього використовується алгоритм стиснення LZW, аналогічний до того, що використовується в TIFF. Однак, варто зазначити, що GIF відрізняється від TIFF декількома нюансами.

По-перше, GIF підтримує анімацію, що дозволяє зберігати кілька зображень в одному файлі.

По-друге, GIF використовує палітру кольорів, що обмежує кількість кольорів, які можуть бути використані в зображенні, до 256 [9]. Саме через цю особливість, за останні роки використання цього формату стали менш привабливими, але незважаючи на це, GIF досі продовжує використовуватись для зображень з обмеженою палітрою кольорів, таких як текст, логотипи або прості ілюстрації, і не дивлячись на його недоліки, є досить ефективним.

Алгоритм LZW в GIF працює так само, як і в TIFF. Він створює словник, який динамічно оновлюється під час процесу стиснення, замінюючи найдовші послідовності символів, які вже є в словнику, на відповідні коди словника. Це дозволяє GIF стискати зображення без втрати якості. Але також варто зазначити, що, цей метод може бути неефективним, якщо повторювані послідовності символів є дуже короткими або дані містять високий рівень шуму.

### 6. Метод стиснення JPEG

Метод стиснення JPEG є форматом зображень, який використовується для зберігання растрових зображень. Однією з ключових особливостей JPEG є можливість стиснення з втратами якості. Для цього використовується алгоритм стиснення, що базується на дискретному косинусному перетворенні (DCT) [10]. DCT є методом стиснення, що використовується в форматі JPEG для зменшення розміру файлу за рахунок втрати якості. Він базується на перетворенні просторових даних зображення в частотні дані, що дозволяє відкинути частотні компоненти, які менш важливі для сприйняття людським оком. Процес стиснення JPEG можна розбити на п'ять основних кроків.

Першим кроком зображення розбивається на блоки розміром 8x8 пікселів, цей процес можна описати формулою 2:

$$N = \frac{W \cdot H}{64} \quad (2)$$

де  $N$  – загальна кількість блоків,  $W$  – ширина зображення,  $H$  – висота зображення.

Далі кольоровий простір зображення перетворюється з RGB в YCbCr, що відокремлює яскравість (Y) від хроматичності (Cb та Cr) [11].

Третім кроком кожен блок 8x8 пікселів перетворює просторові дані координати  $x, y$  в частотні дані (компоненти низьких і високих частот) за допомогою DCT. Матриця результатів DCT розраховується за формулою 3:

$$F = T \cdot B \cdot T^T \quad (3)$$

де  $F$  – матриця результатів DCT,  $B$  – матриця 8x8 пікселів зображення,  $T$  – матриця косинусних перетворень розміром 8x8,  $T^T$  – транспонована матриця перетворень.

Матриця косинусних перетворень розраховується за формулою 4:

$$T_{[i,j]} = \alpha(i) \cdot \cos\left(\frac{(2j+1) \cdot i \cdot \pi}{16}\right) \quad (4)$$

де  $i$  – рядок матриці,  $j$  – стовпець матриці,  $\alpha(i)$  – визначається за формулою 5:

$$\alpha(i) = \begin{cases} \frac{1}{\sqrt{8}}, & \text{якщо } i = 0 \\ \frac{1}{2}, & \text{якщо } i > 0 \end{cases} \quad (5)$$

Четвертим кроком частотні координати кожного блоку нормалізуються за допомогою таблиці квантування, що зменшує точність високочастотних компонентів. На цьому етапі відбувається втрата інформації. Коефіцієнти ДКП квантуються, тобто округляються до заданих значень. Матриця квантування розраховується за формулою 6:

$$B' = \left\lfloor \frac{F}{Q} \right\rfloor \quad (6)$$

де  $B'$  – квантизовані коефіцієнти DCT,  $F$  – матриця результатів DCT,  $Q$  – матриця квантування,  $\lfloor \cdot \rfloor$  – операція взяття підлоги (округлення вниз),

Матриця  $Q$  може бути визначена за допомогою таблиць квантування, які містять значення квантування для різних частотних компонентів та рівнів стиснення [6].

Останнім кроком нормалізовані блоки кодується за допомогою кодування Хаффмана для подальшого зменшення розміру файлу і збираються разом із заголовком JPEG.

Важливо зазначити, що, хоча JPEG є алгоритмом стиснення з втратами, він дозволяє вибрати баланс між розміром файлу та якістю зображення за допомогою параметра якості. Високі значення якості приводять до меншого стиснення та вищої якості зображення, тоді як низькі значення якості приводять до більшого стиснення та нижчої якості зображення.

Оскільки JPEG є одним з найефективніших методів стиснення, переважно для фотографій, які зазвичай містять велику кількість деталей та кольорів, і при цьому зміна якості зображень майже не видна людському оку, він є найпопулярнішим методом стиснення зображень в усьому світі.

### 7. Метод стиснення WEBP

WEBP є форматом зображень, розробленим Google, який активно використовується у веб-застосунках. Однією з ключових особливостей цього формату є можливість стиснення з втратами якості. Для цього використовується алгоритм стиснення VP8.

VP8 є алгоритмом стиснення з втратами, що використовується в форматі WEBP, для зменшення розміру файлу. Він базується на використанні дискретного косинусного перетворення (DCT), аналогічного до того, що використовується в JPEG. DCT перетворює зображення з просторового (піксельного) представлення до частотного. Це дозволяє видалити високо-частотні компоненти, які людське око менш чутливе, зменшуючи розмір файлу без значного зниження якості зображення.

На відміну від JPEG, у форматі WEBP, який використовує алгоритм стиснення VP8, розбивка на блоки може бути більш гнучкою. VP8 може використовувати блоки різного розміру, включаючи 4×4, 8×8, 16×16 та інші. Це дозволяє алгоритму оптимізувати стиснення для різних частин зображення.

Також, варто зазначити, що VP8 відрізняється від JPEG ще декількома нюансами. По-перше, VP8 використовує більш ефективний метод кодування, який включає в себе прогнозування між кадрами та внутрішньо-кадрове прогнозування. По-друге, VP8

включає в себе вбудований механізм корекції помилок, що дозволяє відновити зображення навіть при втраті даних під час передачі. По-третє, VP8 підтримує альфа-канал, що дозволяє створювати прозорі зображення.

Важливо зазначити, що, хоча VP8 є алгоритмом стиснення з втратами, він може бути неефективним для стиснення деяких типів даних. Наприклад він може бути менш ефективним для стиснення зображень з високою деталлю, оскільки він може призвести до втрати дрібних деталей під час стиснення.

Незважаючи на недоліки та переваги кожного з методів стиснення, важливо враховувати багато факторів, таких як якість зображення, розмір файлу, підтримка прозорості та інші вимоги до стиснення.

### 8. Пропонований метод аналізу стиснення зображень для зберігання в децентралізованих блокчейн сховищах

Для вирішення проблеми ефективного зберігання зображень у блокчейн сховищах, пропонується метод аналізу, що ґрунтується на поєднанні технологій стиснення зображень та децентралізованої системи зберігання IPFS (InterPlanetary File System). Це було обрано через актуальність проблеми збереження великих обсягів даних у децентралізованих мережах. Цей метод дозволяє оцінити вплив різних методів стиснення на ефективність зберігання зображень, включаючи основні показники: розмір файлу, якість зображення, час завантаження та пропускну здатність.

З метою реалізації цього методу було розроблено який забезпечує користувачів можливістю імпортувати зображення та вибирати методи стиснення. Після імпорту зображення стискаються за допомогою різних алгоритмів, таких як PNG (deflate), TIFF (lzw), GIF (lzw-like) для стиснення без втрат, та JPEG (DCT), WEBP (VP8) для стиснення з втратами. Цей набір методів було обрано для забезпечення широкого спектру можливих сценаріїв використання.

Якість зображення, оцінюється за допомогою метрики PSNR (Peak Signal-to-Noise Ratio) і зберігається для подальшого аналізу. Використання метрики PSNR було обрано через її здатність об'єктивно оцінювати якість зображення після стиснення, що є критично важливим для цього дослідження.

Стиснені зображення завантажуються в IPFS через сервіс Pinata, який виступає посередником для доступу до всесвітньої мережі. Використовуючи технологію IPFS, кожне завантажене зображення отримує унікальний хеш, що забезпечує його однозначну ідентифікацію та можливість доступу до нього в будь-який момент часу. Це гарантує безпеку та незмінність даних, що є важливим аспектом у контексті децентралізованих мереж.

Після завантаження зображень в IPFS, отримані хеші зберігаються у смарт контракті, розгорнутому в локальній блокчейн мережі, що імітує Ethereum. Для цього використовуються технології Ganache для розгортання локальної мережі та Truffle для управління смарт контрактами, написаними на мові Solidity. Використання цих технологій забезпечує високу надійність та масштабованість системи.

Фронтенд частина веб-застосунку, реалізована за допомогою технології ASP.NET RazorPages, забезпечує інтерактивний і зручний у використанні інтерфейс. Користувачі можуть завантажити зображення зі свого пристрою, вибрати методи стиснення за допомогою чекбоксів, переглянути зібрану інформацію про стиснені зображення, порівняльні графіки, а також відкрити та переглянути всі стиснені зображення для візуального порівняння якості. Інтерфейс користувача розроблений з урахуванням принципів юзабіліті, що забезпечує інтуїтивно зрозуміле взаємодію та мінімізацію помилок з боку користувача. Це було обрано, оскільки зручність використання та інтуїтивність інтерфейсу є критичними для залучення широкої аудиторії користувачів.

Бекенд частина, реалізована на основі .NET 8 та C# 12.0, відповідає за обробку даних, взаємодію з IPFS та блокчейном, а також за виконання основних бізнес-логік застосунку. Використовуючи бібліотеку ImageMagick, бекенд частина здійснює стиснення

зображень за обраними методами, обчислюючи розмір файлу та метрику якості PSNR. Це рішення було обрано через високу продуктивність та гнучкість бібліотеки ImageMagick, яка дозволяє ефективно працювати з різними форматами зображень.

За допомогою HttpClient відбувається завантаження стиснених зображень до IPFS і отримання унікальних хешів для кожного зображення. Для збирання показників часу завантаження в IPFS та пропускної здатності використовуються інструменти вимірювання часу, такі як Stopwatch, для обчислення тривалості завантаження та розміру файлу після стиснення. Це забезпечує точний та надійний збір даних для подальшого аналізу.

Блокчейн частина системи відповідає за збереження хешів зображень у децентралізованій мережі, що гарантує їх безпеку, незмінність та прозорість. Локальна блокчейн мережа розгортається за допомогою Ganache, що дозволяє швидко створити середовище для тестування, а Truffle використовується для управління смарт контрактами, написаними на мові Solidity. Цей підхід був обраний через його надійність та широке визнання в спільноті розробників блокчейн-рішень.

Нижче наведено малюнок який візуалізує взаємодію вищезазначених модулів розробленої системи (див. рис. 4).

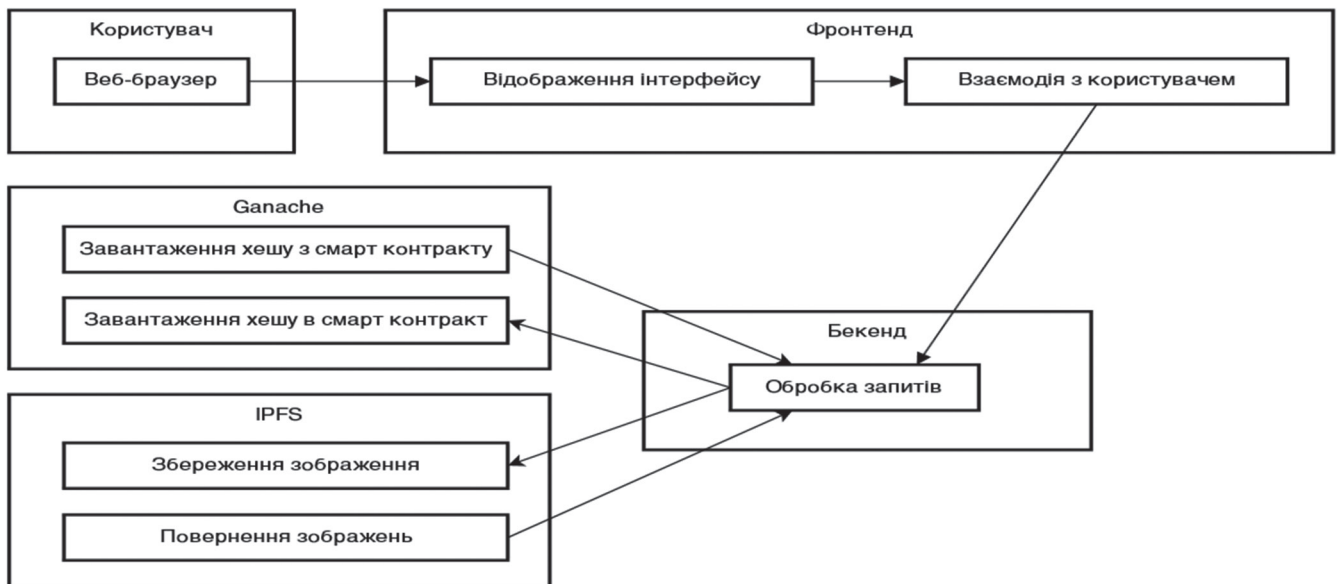


Рис. 4. Архітектура пропонованого застосунку для аналізу.

Таким чином, розроблений метод дозволяє користувачам проводити комплексний аналіз ефективності різних методів стиснення зображень у контексті зберігання в децентралізованих блокчейн сховищах. Це забезпечує можливість знайти оптимальний баланс між розміром файлу та якістю зображення, що

є критично важливим для багатьох сучасних застосунків. Завдяки використанню передових технологій та інтеграції з блокчейн, пропонований метод забезпечує високу надійність, прозорість та масштабованість, що робить його інноваційним рішенням у сфері управління цифровими даними.

## 9. Експеримент

Для дослідження ефективності різних методів стиснення зображень у контексті їх зберігання та передачі в IPFS було обрано різноманітний набір зображень. Зображення представляють типові сценарії використання у веб-застосунках і включають фотографії, медичні зображення, графіку та логотипи. Всі зображення були отримані у форматі BMP, що забезпечує високу якість без втрат і є оптимальним для подальшого порівняльного аналізу методів стиснення.

Для кожного зображення було застосовано п'ять популярних методів стиснення, такі як: JPEG, PNG, WEBP, GIF та TIFF.

JPEG використовується для фотографій завдяки своєму ефективному алгоритму стиснення з втратами, який добре працює з деталями та градієнтами кольору. PNG забезпечує стиснення без втрат, що робить його ідеальним для графіки та ілюстрацій. WEBP поєднує в собі переваги як стиснення з втратами, так і без втрат, пропонуючи високий ступінь компресії. GIF часто використовується для простих графічних зображень і анімацій, тоді як TIFF забезпечує високоякісне стиснення без втрат для збереження максимальної якості.

Набір досліджуваних даних включає в себе 20 фотографій, які відрізняються такими характеристиками як: наявність монохромності, кольорова гамма та температура, контрастність, насиченість та різна щільність пікселів.

Також набір даних містить 15 логотипів, які можна розділити на 3 групи по 5 логотипів: кольорові логотипи, чорно-білі логотипи та чорно-білі логотипи без заливки кольором.

Окрім цього датасет включає в себе 5 графічних зображень, серед яких є чорно-білі зображення. Графічні зображення відрізняються за розміром, кольором, насиченістю та іншими характеристиками.

Також для всебічного аналізу, в досліджувані дані було додано чорно-білі медичні зображення, такі як знімки МРТ, зображення клітин та інші зображення з медичної сфери.

Кожне зображення з набору даних було ретельно підібране, щоб включати широкий спектр характеристик. Першою з основних характеристик при відборі зображень була роздільна здатність, тобто кількість пікселів по горизонталі та вертикалі, яка впливає на розмір файлу та якість зображення. Також пріоритетною характеристикою була щільність пікселів, що визначається кількістю пікселів на дюйм і впливає на чіткість зображення. При відборі також була важлива різноманітність кольорової гамми, яка відображає діапазон кольорів, що наявні на зображенні і контрастність та насиченість.

Після того як був сформований набір даних, стало можливим провести аналіз досліджуваних зображень,

для того щоб визначити оптимальний метод стиснення для зберігання у блокчейн сховищі. Для використання показники що описані нижче.

Оскільки одною з основних цілей під час зберігання файлів є зниження витрат на зберігання, то було визначено один з головних показників, який прямо впливає на вартість зберігання зображень в IPFS – розмір завантаженого файлу. Відсоткове співвідношення розміру стисненого файлу до початкового файлу обчислювалось за формулою 7:

$$S = \frac{C}{O} \cdot 100 \quad (7)$$

де  $S$  – відсоткове співвідношення розмірів,  $C$  – розмір стиснутого файлу,  $O$  – розмір оригінального файлу.

Аналогічно до розміру файлу, на вартість зберігання також впливає витрачена пропускна здатність каналу (мережі) який був наданий кінцевому користувачу. Пропускна здатність, вимірювана у кілобайтах на секунду обчислювалась за формулою 8:

$$Tp = \frac{C}{t} \quad (8)$$

де  $Tp$  – пропускна здатність,  $C$  – розмір стиснутого файлу у кілобайтах,  $t$  – час, витрачений на завантаження в IPFS, вимірюваний в секундах.

Також, важливою вимогою для клієнта, під час процесу стиснення є максимально можливе збереження якості файлу після форматування. Якість вимірюється шляхом калькуляції піксельної різниці сигналу до шуму (PSNR) розраховувалась за формулою 9:

$$psnr = 10 \cdot \log_{10} \left( \frac{255^2}{mse} \right) \quad (9)$$

де  $psnr$  – пікове відношення сигналу до шуму,  $mse$  – значення середньоквадратичної похибки.

Ці показники були обрані для об'єктивної оцінки ефективності кожного методу стиснення з точки зору розміру файлу, швидкості завантаження, та якості зображення.

В першу чергу, було виміряно та проаналізовано розміри фотографій після стиснення за допомогою різних методів (див. рис. 5).

Як видно з діаграми, найбільший розмір файлу після стиснення спостерігався для методу TIFF (алгоритм LZW). Це можна пояснити тим, що методи стиснення без втрат зазвичай зберігають більше інформації, ніж методи з втратами. З іншого боку, метод WEBP (алгоритм VP8) показав найкращий результат з точки зору зменшення розміру файлу.

Другим важливим показником було вимірювання якості зображення після стиснення за допомогою метрики PSNR (Peak Signal-to-Noise Ratio) (див. рис. 6).

Як показує діаграма, методи стиснення без втрат, такі як PNG і TIFF, повністю зберігають якість зображення, що підтверджується максимальними



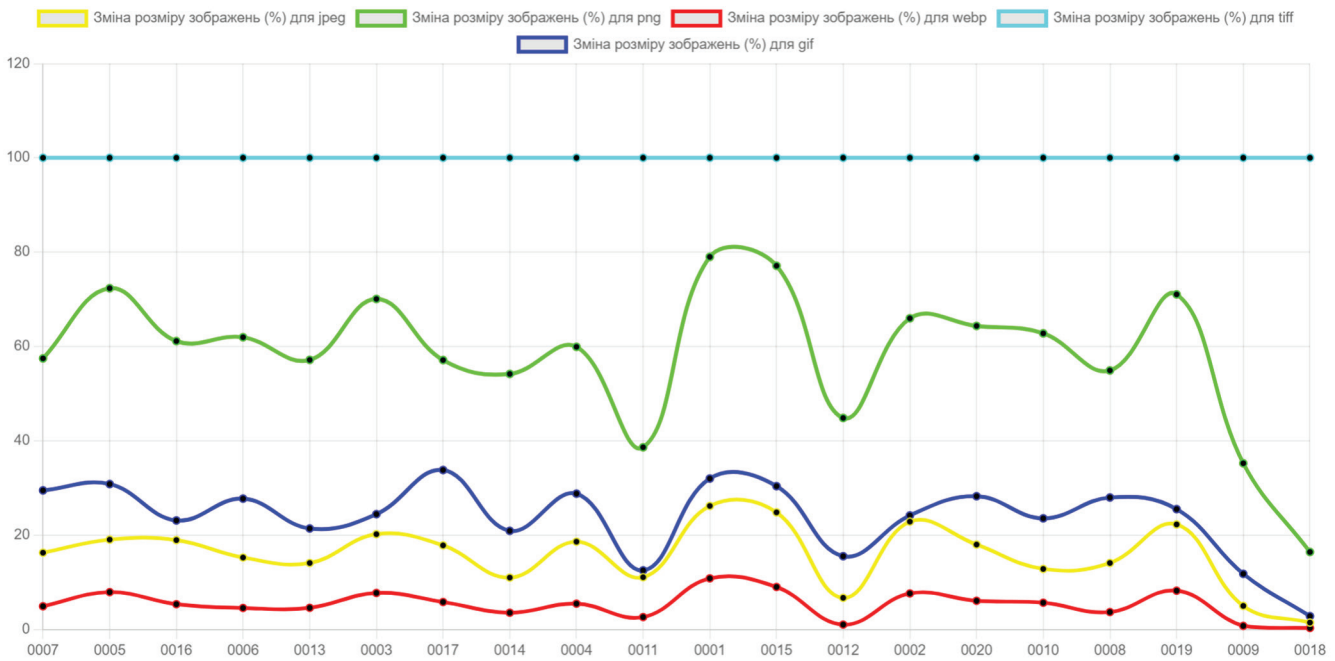


Рис. 5. Графік зміни розміру для фотографій

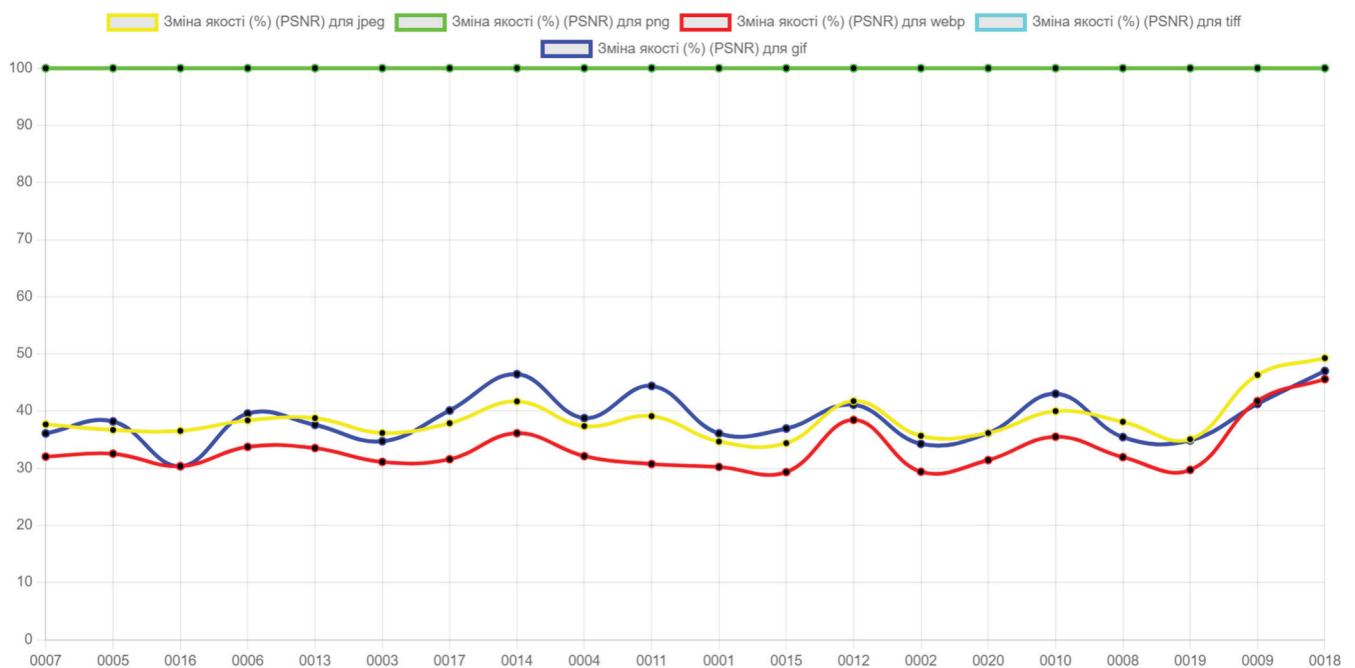


Рис. 6. Графік зміни якості для фотографій

значеннями PSNR. Методи стиснення з втратами, такі як JPEG, WEBP і GIF, демонструють нижчі значення PSNR, що є очікуваним результатом через втрату частини інформації при стисненні.

Наступним важливим показником була вимірювана пропускна здатність при завантаженні зображень в IPFS (див. рис. 7).

Як показує діаграма, найбільш ефективні з точки зору пропускної здатності, тобто витрачають її менше, є методи стиснення з втратами, такі як WEBP, JPEG та GIF. Це можна пояснити тим, що менші розміри файлів дозволяють швидше передавати дані через мережу.

Аналіз показав, що для фотографій, які зазвичай містять велику кількість деталей та кольорів, методи стиснення з втратами, такі як JPEG і WEBP, є більш ефективними для зменшення розміру файлу. На основі результатів ми робимо висновок, що метод JPEG забезпечує оптимальний баланс між зменшенням розміру файлу та якістю зображення для стиснення фотографій.

Аналогічно фотографіям, було виміряно та проаналізовано розміри медичних зображень після стиснення за допомогою різних методів (див. рис. 8).

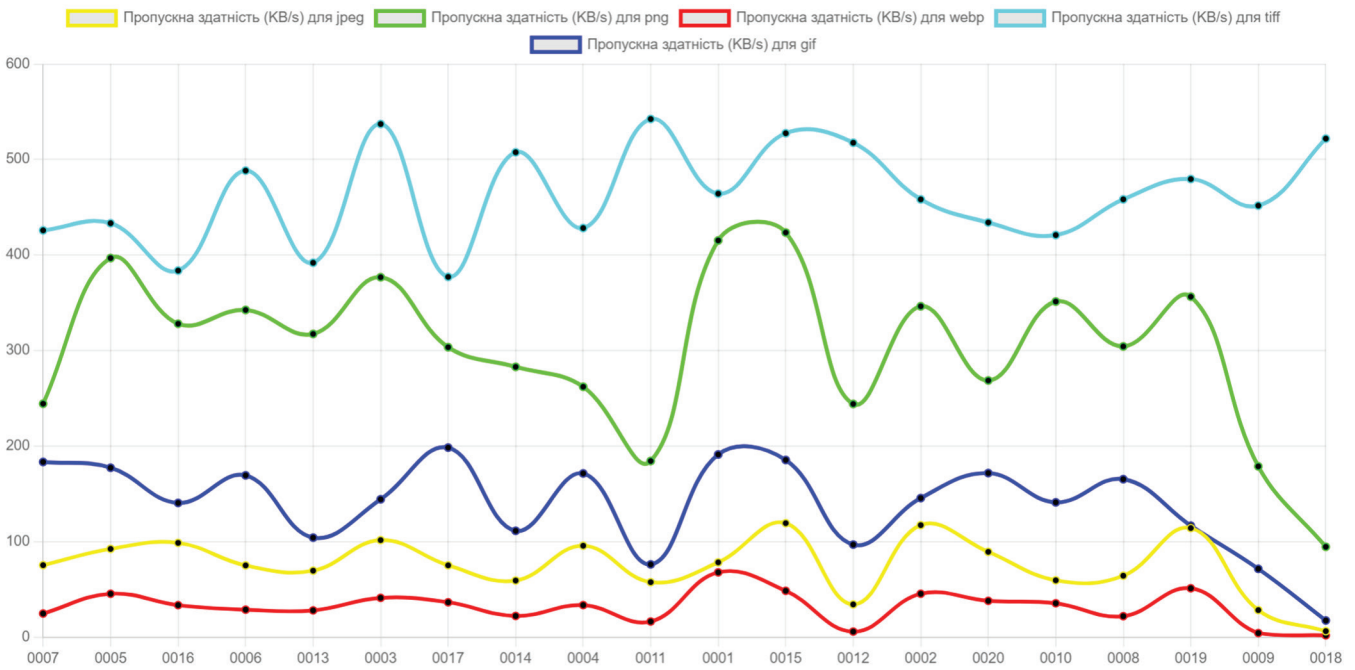


Рис. 7. Графік пропускної здатності для фотографій

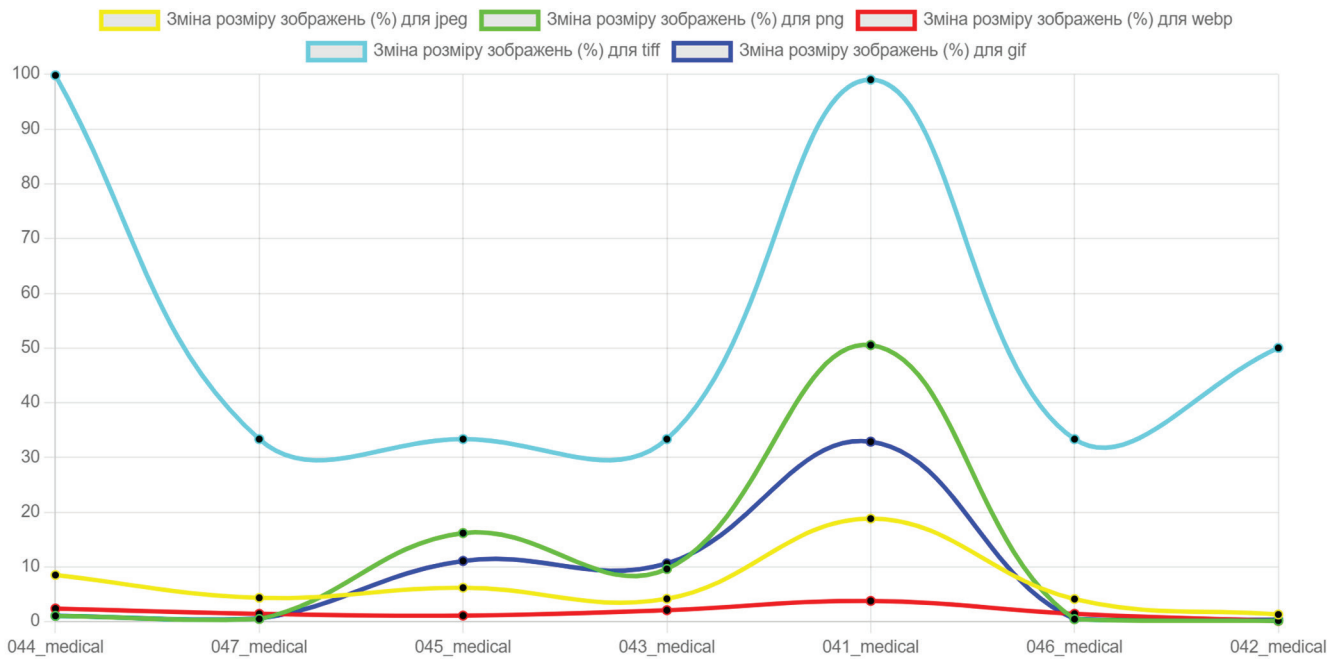


Рис. 8. Графік зміни розміру для медичних зображень

Як видно з діаграми, методи стиснення з втратами, такі як JPEG та WEBP, значно зменшили розмір файлу, пропонуючи ефективніші результати з точки зору обсягу даних. Також можна побачити що метод PNG, який стискає зображення без втрат, показав найкращі результати для більшості даних, в порівнянні з іншими методами стиснення, навіть з тими, що нехтують якістю під час форматування. Другим важливим показником було вимірювання якості медичних зображень після стиснення за допомогою метрики PSNR (див. рис. 9).

Як показує діаграма, методи стиснення без втрат, такі як PNG і TIFF, повністю зберігають якість зображення. Методи стиснення з втратами, такі як JPEG, WEBP та для більшості вибірки GIF, демонструють нижчі значення PSNR, що може бути критичним для медичних зображень, де важливе збереження максимальної кількості деталей.

Наступним важливим показником була виміряна пропускна здатність при завантаженні медичних зображень в IPFS (див. рис. 10).

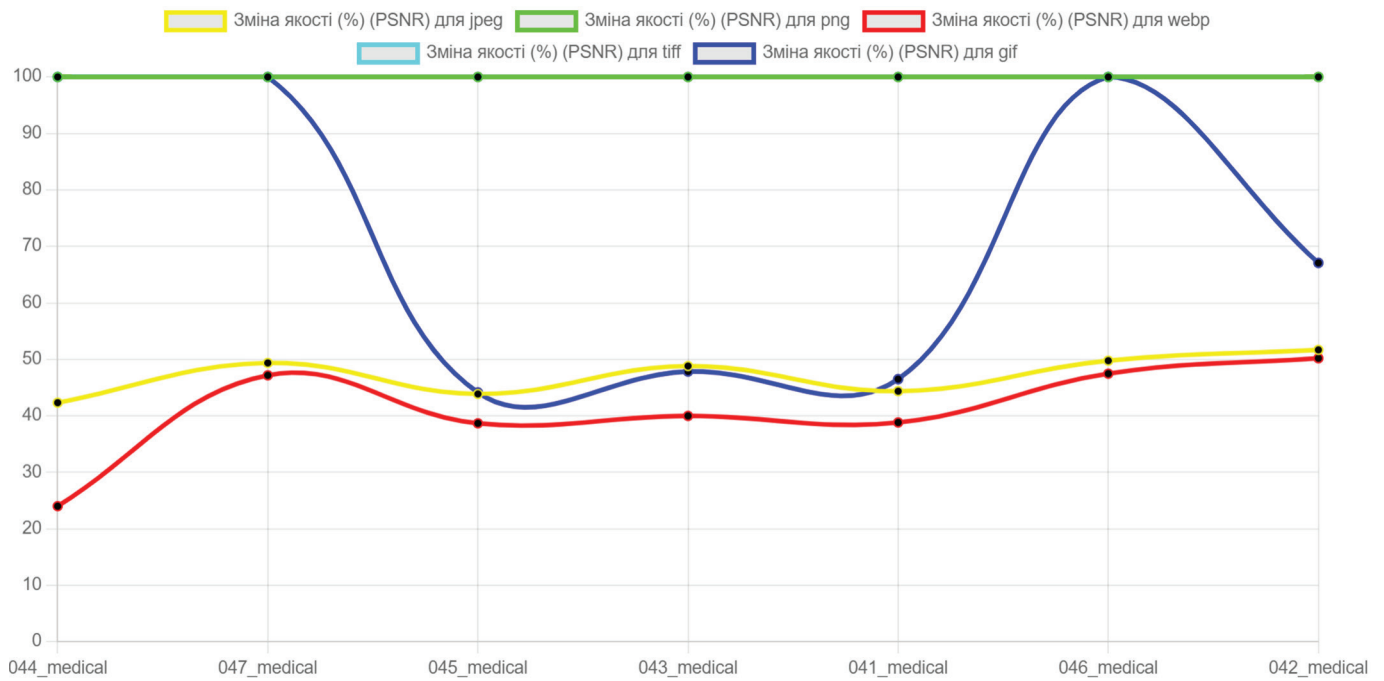


Рис. 9. Графік зміни якості для медичних зображень

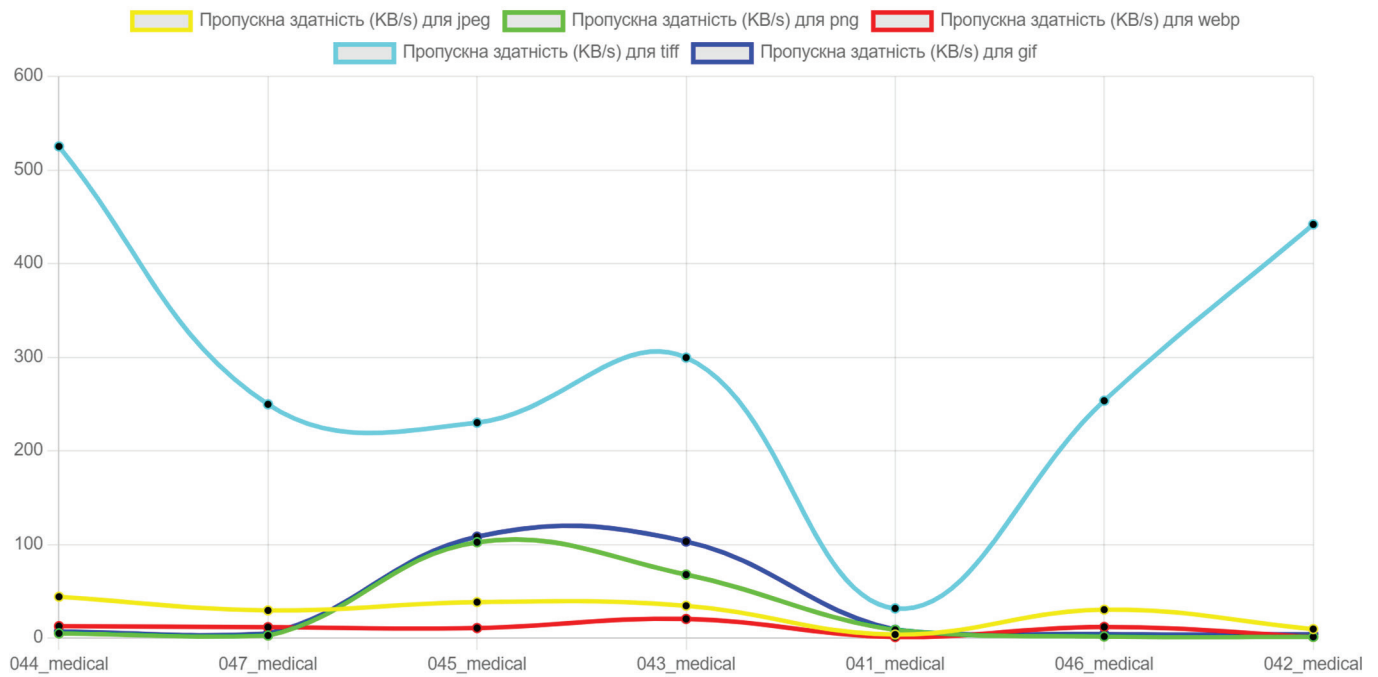


Рис. 10. Графік пропускну здатності для медичних зображень

Як показує діаграма, методи стиснення з втратами, такі як WEBP та JPEG, є найбільш ефективними з точки зору пропускну здатності, але в той же час, PNG показав хоч і не для всієї вибірки, але для більшості, найкращі показники.

Аналіз показав, що для медичних зображень критично важливим є збереження максимальної якості, тому метод PNG є більш придатним за інші, оскільки стискає зображення без втрати якості і показує найкращі результати для більшості досліджуваних файлів за такими характеристиками як розмір файлу

та пропускну здатність під час завантаження в блокчейн сховище. Проте, якщо необхідно надійно зменшити розмір файлу в декілька разів для усієї вибірки, враховуючи можливу втрату важливих деталей, може бути використаний метод стиснення з втратами WEBP, оскільки показав найкращі результати для цього показника.

Далі було виміряно та проаналізовано розміри графіки та логотипів після стиснення за допомогою різних методів (див. рис. 11).

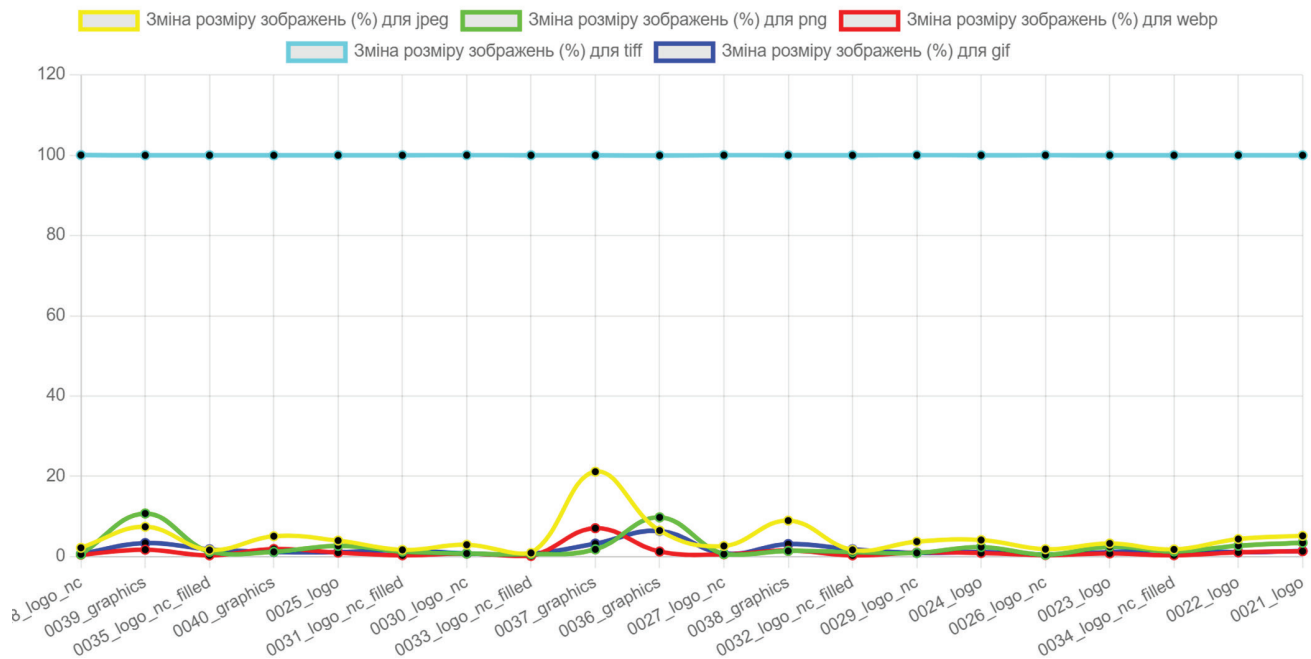


Рис. 11. Графік зміни розміру графічних зображень і логотипів

Як видно з діаграми, метод стиснення без втрат TIFF, має більші розміри файлів у порівнянні з методами з втратами, такими як JPEG, WEBP та GIF.

У той же час метод стиснення без втрат PNG показує кращі результати стиснення за JPEG та GIF (див. рис. 12.)

Зображення 1				
Type: png, Quality: 100%, Size: 7.51 KB	Time: 1.22s,	Throughput: 6.18 KB/s	<a href="#">Open Image</a>	
Type: webp, Quality: 42.74%, Size: 8.17 KB	Time: 0.93s,	Throughput: 8.78 KB/s	<a href="#">Open Image</a>	
Зображення 2				
Type: png, Quality: 100%, Size: 35.35 KB	Time: 1.27s,	Throughput: 27.76 KB/s	<a href="#">Open Image</a>	
Type: webp, Quality: 30.72%, Size: 38.54 KB	Time: 1.23s,	Throughput: 31.32 KB/s	<a href="#">Open Image</a>	
Зображення 3				
Type: png, Quality: 100%, Size: 180.47 KB	Time: 1.33s,	Throughput: 135.24 KB/s	<a href="#">Open Image</a>	
Type: webp, Quality: 25.3%, Size: 706.89 KB	Time: 1.47s,	Throughput: 481.33 KB/s	<a href="#">Open Image</a>	

Рис. 12. Порівняння розмірів деяких файлів для WEBP та PNG

Попередній аналіз показав, що пропускна здатність прямо залежить від розміру файлу, тому проводити додатковий аналіз діаграми пропускної здатності для графічних зображень та логотипів не є необхідним. Також оскільки для цих зображень є доречним максимальне збереження, тому метод PNG є оптимальнішим за інші, оскільки проводить безвтратне стискання і показує гарні результати стиснення файлу. Проте, як альтернативу для методу PNG при збереженні графічних файлів можна відмітити WEBP, оскільки цей метод стиснення показує чи найкращі показники співвідношення якості та розміру файлу серед методів стиснення з втратами. Проте в такому випадку буде мати місце значна втрата якості.

### Висновки

В ході проведення аналізу було досліджено методи стиснення зображень для зберігання у децентралізованих блокчейн сховищах на прикладі IPFS.

Для цього спочатку було розглянуто особливості зберігання даних у блокчейн сховищах, зокрема в IPFS і проаналізовано методи стиснення зображень, які можуть бути застосовані для зменшення розміру файлів перед зберіганням в IPFS.

Наступним кроком було визначено показники ефективності для кожного з методів стиснення (розмір файлу, якість зображення, пропускна здатність при завантаженні, час завантаження).

Після цього для проведення дослідження було розроблено бекенд на .NET 8 та фронтенд на Razor Pages для імпорту та стиснення зображень. Для імітації роботи реальної блокчейн мережі було розгорнуто локальну блокчейн мережу Ethereum за допомогою Ganache й написано смарт контракт на мові Solidity [12] який був пізніше задеплоєний в локальну блокчейн мережу за допомогою Truffle.

Після реалізації системи, яка дозволить провести аналіз, було проведено експеримент зі стискання зображень різними методами та збереженням їх в IPFS із фіксацією отриманих хешів у смарт-контрактах на локальній блокчейн мережі. За результатами експерименту зібрано та проаналізовано дані про розмір файлів, якість зображень (за метрикою PSNR), час завантаження в IPFS та пропускну здатність для кожного методу стиснення;

Для повноти картини було візуалізовано результати дослідження у вигляді графіків та таблиць.

За результатами дослідження, по-перше, було визначено, що методи стиснення без втрат, такі як PNG і TIFF, дозволяють зберегти високу якість зображення. Проте, це призводить до значного збільшення розміру файлу. Така особливість може бути критично важливою для типів зображень, де максимальна деталізація є необхідною, як-от медичні зображення.

По-друге, дослідження показало, що методи стиснення з втратами, такі як JPEG та WEBP, дозволяють значно зменшити розмір файлу. Це покращує ефективність зберігання та передачі даних у блокчейн сховищах. Зокрема, вони є оптимальними для використання з фотографіями та графікою, де допускається деяке зниження якості без значних втрат візуальної інформації.

Крім того, було встановлено, що метод WEBP демонструє найкращі результати з точки зору зменшення розміру файлу при збереженні достатньої якості зображення. Це робить його оптимальним для використання у випадках, коли необхідно знайти баланс між розміром файлу та якістю.

Дослідження також виявило, що пропускну здатність при завантаженні в IPFS є найвищою у методів стиснення з втратами, що зумовлено їх меншим розміром файлів. Це сприяє швидшій передачі даних у мережі, що є важливим показником ефективності.

На основі отриманих результатів можна зробити висновок, що для випадків, де критично важлива висока якість зображень, найкращим вибором є метод стиснення без втрат PNG. Для більш ефективного використання ресурсів та зменшення часу завантаження, методи стиснення з втратами, такі як JPEG та WEBP, є більш доцільними. Проведене дослідження

показало, що метод JPEG забезпечує оптимальний баланс між зменшенням розміру файлу та якістю зображення.

Для підтвердження отриманих результатів та зроблених висновків необхідні подальші дослідження. Вони мають охоплювати різноманітні типи зображень та включати більш складні сценарії використання, що дозволить отримати більш точні дані щодо ефективності кожного методу стиснення у різних умовах.

#### Список літератури:

- [1] Blockchain Technology Explained: Powering Bitcoin. URL: <https://www.ibm.com/topics/what-is-blockchain> (date of access: 29.03.2023)
- [2] Терещенко Г.Ю., Груздо І.В. Застосування симетричних алгоритмів в блокчейні. Біоніка інтелекту. – Харків : ХНУРЕ. – 2020. – № 1 (94). – С. 33-39
- [3] Huckle S. Internet of Things, Blockchain and Shared Economy Applications / S. Huckle, R. Bhattacharya, M. White, N. Beloff // Procedia Comput. Science. – Oct. 2016. – Vol. 98. – P. 461–466.
- [4] W. Pennebaker, J. Mitchell, JPEG: Still Image Data Compression Standard. – Springer New York, NY, 1992. – 638p. – ISBN: 978-0-442-01272-4.
- [5] 6. N. Bilous, G. Tereshchenko Copyright protection using blockchain. Біоніка інтелекту – Харків : ХНУРЕ. – 2019. – № 1 (92). – С. 52-58.
- [6] GIF File Format Summary. URL: <https://www.fileformat.info/format/gif/egff.htm> (date of access: 07.04.2023).
- [7] LZW compression. URL: <https://www.techtarget.com/whatis/definition/LZW-compression> (date of access: 07.04.2023).
- [8] Білоконенко В.М , Ревенчук І.А. Алгоритми сегментації зображень на базі побудови матриць збігів. Східно-Європейський журнал передових технологій.- №2(62), том 2.-2013.-С.43-45
- [9] Специфікація алгоритму стискання даних формату DEFLATE версія 1.3 (RFC1951). URL : <http://www.kytk.org.ua/post/cpetsyifikatsiya-alhorytmu-styskannya-danykh-formatu-deflate-versiya-1-3-rfc1951> (date of access: 07.04.2023).
- [10] J. Solem. Programming Computer Vision with Python: Tools and algorithms for analyzing images. O'Reilly Media, Inc., 2012. 264p
- [11] JPEG YCbCr Support. URL: <https://learn.microsoft.com/en-us/windows/win32/wic/jpeg-ycbcr-support> (date of access: 08.04.2023).
- [12] Шаронова Н.В., Терещенко Г.Ю. Проблеми і перспективи практичного застосування інформаційної технології blockchain в smart-контрактах. Інтелектуальні системи та інформаційні технології (ISIT-2019). – Матеріали Міжн. Наук.-практ. Конф. – Одеса, 19–24 серпня 2019 р. – С. 214–219.

*Надійшла до редколегії 7.02.2024*