

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматизації та інтелектуальних інформаційних технологій

(повна назва кафедри (предметної, циклової комісії))

### Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: **Модифікація алгоритму стиснення зображень JPEG на основі**

**вейвлет-перетворення**

Виконав: студент 2 курсу, групи ІСТ-21м  
спеціальності

126 Інформаційні системи та технології

(шифр і назва спеціальності)

Конопльов Е.Ю.

(прізвище та ініціали)

Керівник д.т.н., професор Бісікало О.В.

(прізвище та ініціали)

Рецензент д.т.н., професор Ковтун В.В.

(прізвище та ініціали)

Допущено до захисту

Завідувач кафедри АІТ

д.т.н., проф. Бісікало О.В.

(прізвище та ініціали)

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
Освітньо-кваліфікаційний рівень магістр  
Освітньо-професійна програма «Інформаційні технології аналізу даних та зображень»  
Спеціальність 126 – «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ  
Завідувач кафедри АІТ  
д.т.н. проф. О.В. Бісікало  
" \_\_\_ " \_\_\_\_\_ 2022 року

**ЗАВДАННЯ  
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Конопльову Едуарду Юрійовичу

1. Тема магістерської кваліфікаційної роботи Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення

керівник магістерської кваліфікаційної роботи Бісікало Олег Володимирович,  
д.т.н., професор каф. АІТ

затверджена наказом вищого навчального закладу від « \_\_\_ » \_\_\_\_\_ 2022 року № \_\_\_\_\_

2. Строк подання студентом магістерської кваліфікаційної роботи «07» грудня 2022 р.

3. Вихідні дані до магістерської кваліфікаційної роботи Тип цифрового зображення – півтонові та кольорові; мінімальна роздільна здатність зображення 128 x 128 пікселів, максимальна роздільна здатність 2048 x 2048 пікселів; коефіцієнт стиснення від 3% до 20%.

4. Зміст розрахунково-пояснювальної записки: 1. Аналіз сучасних методів стиснення зображень. 2. Застосування ортогональних перетворень для стиснення зображень. 3. Розробка та реалізація модифікованого алгоритму JPEG. 4. Економічний розділ. Додатки.

5. Перелік графічного матеріалу: 1. Основні етапи стиснення цифрових зображень. 2. Схема стиснення зображення по стандарту JPEG. 3. Стиснення зображення. Схема програми. 4. Результат роботи програми за середньоквадратичним відхиленням значень пікселів стиснутого зображення від оригіналу. 5. Результат роботи програми за критерієм якості PSNR. 6. Результати роботи програми.

## 6. Консультанти розділів проекту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|---|----------------|------------------|
|        |   | завдання видав | завдання прийняв |
| 1-3    | Гармаш В.В., к.т.н., доцент каф. АІТ      |                |                  |
| 4      | Лесько О.Й., к.е.н., професор каф. ЕПВМ   |                |                  |
|        |   |                |                  |

7. Дата видачі завдання \_\_\_\_\_ «    » \_\_\_\_\_ 2022 року

## КАЛЕНДАРНИЙ ПЛАН

| N з/п | Назва етапів дипломного проекту                                | Строк виконання етапів проекту | Примітка |
|-------|--|--------------------------------|----------|
| 1     | Аналіз сучасних методів стиснення зображень                    | 20.10.2022 р.                  |          |
| 2     | Застосування ортогональних перетворень для стиснення зображень | 01.11.2022 р.                  |          |
| 3     | Розробка та реалізація модифікованого алгоритму JPEG           | 15.11.2022 р.                  |          |
| 4     | Економічний розділ   | 20.11.2022 р.                  |          |
| 5     | Оформлення пояснювальної записки і графічного матеріалу        | 30.11.2022 р.                  |          |
| 6     | Попередній захист  | 07.12.2022 р.                  |          |
| 7     | Остаточний захист  |                                |          |

Студент

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
Конопльов Е.Ю.  
(прізвище та ініціали)

Керівник роботи (проекту)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
Бісікало О.В.  
(прізвище та ініціали)

## РЕФЕРАТ

У магістерській кваліфікаційній роботі «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» проведено аналіз існуючих методів стиснення цифрових зображень, розглянуто можливість модифікації алгоритму JPEG для стиснення інформації та його переваги. Для стандартного алгоритму JPEG з постійною таблицею Хаффмана побудована модифікація. Вона має таку ж трудомісткість, але дає вигоду в стисненні до 20%. Розроблено алгоритмічне та програмне забезпечення, яке демонструє переваги запропонованого методу.

## **ABSTRACT**

In the master's qualification work "Modification of the JPEG image compression algorithm based on wavelet transformation" an analysis of existing digital image compression methods was carried out, the possibility of modifying the JPEG algorithm for information compression and its advantages were considered. A modification is built for the standard JPEG algorithm with a constant Huffman table. It has the same labor intensity but gives a gain in compression of up to 20%. Algorithmic and software have been developed that demonstrate the advantages of the proposed method.

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....  | 9  |
| ВСТУП.....  | 9  |
| РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ ЗОБРАЖЕНЬ                            | 11 |
| 1.1 Загальна характеристика задачі .....  | 11 |
| 1.2 Представлення цифрових зображень .....                                      | 12 |
| 1.3 Класифікація методів стиснення. Основні характеристики .....                | 14 |
| 1.4 Алгоритми стиснення зображень без втрат .....                               | 17 |
| 1.4.1 Алгоритм RLE .....  | 17 |
| 1.4.2. Стиснення за методом Хаффмана.....                                       | 18 |
| 1.4.3 Алгоритм Лемпеля-Зива (LZ-compression) .....                              | 19 |
| 1.4.4 Алгоритм Лемпеля-Зива-Велч (Lempel-Ziv-Welch -LZW) .....                  | 20 |
| 1.5 Алгоритми стиснення з втратами .....  | 21 |
| 1.5.1 Стиснення за стандартом JPEG .....  | 21 |
| 1.5.2 Фрактальне стиснення зображень .....                                      | 22 |
| 1.5.3 Метод усіченого блочного кодування .....                                  | 30 |
| 1.6 Вибір і обґрунтування аналогу .....   | 32 |
| 1.7 Вибір напрямку та задач дослідження .....                                   | 32 |
| РОЗДІЛ 2 ЗАСТОСУВАННЯ ОРТОГОНАЛЬНИХ ПЕРЕТВОРЕНЬ ДЛЯ<br>СТИСНЕННЯ ЗОБРАЖЕНЬ..... | 34 |
| 2.1 Перетворення Адамара.....   | 34 |
| 2.2 Перетворення Карунена-Лоева .....   | 37 |
| 2.3 Дискретне косинусне перетворення .....                                      | 39 |
| 2.4 Дискретне вейвлет-перетворення .....  | 41 |
| 2.5 Висновки до розділу .....   | 43 |
| РОЗДІЛ 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДИФІКОВАНОГО<br>АЛГОРИТМУ JPEG .....          | 45 |
| 3.1 Модифікований метод вейвлет перетворення .....                              | 45 |

|  |    |
|--|----|
| 3.2 Модифікований алгоритм JPG .....   | 49 |
| 3.3 Обґрунтування вибору мови програмування.....   | 51 |
| 3.4 Методика порівняння .....  | 53 |
| 3.5 Аналіз результатів роботи модифікованого алгоритму .....   | 56 |
| 3.6 Висновок до розділу .....  | 60 |
| 4 ЕКОНОМІЧНА ЧАСТИНА .....   | 61 |
| 4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки .....  | 61 |
| 4.2 Розрахунок узагальненого коефіцієнта якості розробки.....  | 62 |
| 4.3 Розрахунок витрат на проведення науково-дослідної роботи .....   | 64 |
| 4.3.1 Витрати на оплату праці .....  | 64 |
| 4.3.2 Відрахування на соціальні заходи .....   | 68 |
| 4.3.3 Сировина та матеріали .....  | 68 |
| 4.3.4 Розрахунок витрат на комплектуючі .....  | 70 |
| 4.3.5 Спецустаткування для наукових (експериментальних) робіт .....  | 70 |
| 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт .....  | 71 |
| 4.3.7 Амортизація обладнання, програмних засобів та приміщень.....   | 72 |
| 4.3.8 Паливо та енергія для науково-виробничих цілей .....   | 73 |
| 4.3.9 Службові відрядження .....   | 75 |
| 4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації .....                                    | 75 |
| 4.3.11 Інші витрати .....  | 76 |
| 4.3.12 Накладні (загальновиробничі) витрати .....  | 76 |
| 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором..... | 78 |
| 4.5 Висновки до розділу .....  | 82 |
| ВИСНОВКИ.....  | 84 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....   | 85 |
| ДОДАТКИ.....   | 89 |

|  |     |
|--|-----|
| Додаток А (обов'язковий) – Технічне завдання .....                                       | 90  |
| Додаток Б (обов'язковий) – Графічна частина .....  | 94  |
| Додаток В (обов'язковий) Лістинг модуля програми.....                                    | 101 |
| Додаток Г (обов'язковий) Протокол перевірки навчальної<br>(кваліфікаційної) роботи ..... | 110 |



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВП – вейвлет-перетворення

ДВП – дискретне вейвлет-перетворення

ЦОЗ – цифрова обробка зображень

ЦЗ – цифрове зображення

EZW (Embedded Zerotree Wavelet coder) – алгоритмом вкладеного нульдеревця

JPEG (Joint Photographic Experts Group) – об'єднана група експертів по фотографії

MSE (mean squared error) – середньоквадратична помилка

PSNR (peak signal-to-noise ratio) – пікове відношення сигналу до шуму

## ВСТУП

### **Актуальність теми**

На сьогоднішній день одним з важливих факторів частого відвідування сайту користувачами є швидкість його завантаження. Особливу увагу слід приділити зображенням, оскільки вони складають основну частину контенту. Зображення повинні бути достатньо високої якості, але в той же час мати невеликий розмір файлу. Вимога до розміру файлу виникає із сучасних тенденцій серед користувачів — все частіше використовується мобільний інтернет, який має малу швидкість. Розвиток сучасних інформаційних систем і мереж привів до широкого використання цифрових зображень. У наш час багато галузей техніки, які мають відношення до отримання, обробки, зберігання і передачі інформації, значною мірою орієнтуються на розвиток систем, в яких інформація представлена у вигляді зображень.

Стиснення зображень важливе для підвищення ефективності використання комунікаційних й інформаційно-обчислювальних ресурсів різноманітних систем.

Методам обробки цифрових зображень приділено багато уваги у роботах вітчизняних та зарубіжних вчених [1-26].

### **Мета і задачі дослідження**

Метою даної роботи є підвищення коефіцієнту стиснення цифрових зображень шляхом модифікації алгоритму стиснення JPEG, що дозволяє ефективно стискати зображення при мінімальному спотворенні.

Задачі дослідження можна сформулювати наступним чином:

- Провести аналіз існуючих методів стиснення зображень.
- Розробити модифікований метод стиснення на основі алгоритму JPEG.
- Провести експериментальне дослідження розробленого методу.
- Розробити алгоритмічне та програмне забезпечення.

**Об'єкт дослідження** – процес стиснення цифрових зображень.

**Предмет дослідження** – алгоритм JPEG для стиснення цифрових зображень.

### **Наукова новизна одержаних результатів**

**Удосконалено** метод стиснення цифрових зображень JPEG шляхом його модифікації за рахунок вейвлет-перетворення, що дозволило отримати вииграш у стисненні до 20% порівняно з базовим кодеком JPEG залежно від необхідної якості та вихідного розміру зображення. Модифікація дає вииграш у стисненні як на зображеннях з дрібними деталями, так і на фонових зображеннях.

### **Практичне значення одержаних результатів**

Практичне значення одержаних у роботі результатів полягає в тому, що на основі запропонованого у роботі методу, розроблено та програмно реалізований алгоритм стиснення цифрових зображень, який дозволив підвищити ступінь стискання зображень та зберегти прийнятну якість. Запропонований метод може використовуватись для пришвидшення завантаження сайтів у мережі Інтернет.

## РОЗДІЛ 1

### АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ ЗОБРАЖЕНЬ

#### 1.1 Загальна характеристика задачі

В задачах обробки, передачі та зберігання зображень велике значення мають методи стиснення даних. Як правило дані в таких системах складаються з інтенсивних цифрових потоків. Тому навіть незначне скорочення потоку інформації дозволить збільшити обсяг передачі та зберігання даних. В такому випадку на ефективності застосування систем безпосередньо позначається якість роботи алгоритмів стиснення. При цьому основна частина інформації, яка оброблюється припадає на відеопослідовності та зображення. У зв'язку з цим вельми важливою представляється задача аналізу існуючих методів стиснення зображень [4].

Стиснення переданих даних із їх подальшим відновленням на приймаючій стороні є ефективним способом скорочення інтенсивності цифрового потоку. Як правило, відомі методи стиснення без втрат показують низьку ефективність при роботі з зображеннями. Тому доцільно застосовувати методи стиснення із втратою інформації. Ці методи дозволяють досягати більш високих коефіцієнтів стиснення при цьому зберігаючи гарну якість відновлення.

Можна виділити алгоритми кодування, на основі проведеного аналізу робіт по стисненню зображень, орієнтовані на перетворення стаціонарних і нестаціонарних сигналів. Найбільше поширення отримав метод стиснення на основі дискретного косинусного перетворення для стаціонарних сигналів. Він демонструє ефективні результати для кодування квазіперіодичних сигналів. Опис нестаціонарних сигналів доцільно виконувати за допомогою вейвлет-перетворення, що дозволяє отримати високі коефіцієнти стиснення для сигналів, які описуються кусково-поліноміальними функціями. Узагальнення методів

Фур'є та вейвлет-перетворень для обробки двовимірних сигналів призводить до відомих розділимих перетворень. Вони володіють вираженою анізотропією, пов'язаною з штучно введеними координатними осями при побудові розділимих алгоритмів кодування. Однак просторові властивості більшості реальних зображень ізотропні, що характерно для розділимих породжуючих фільтрів. У зв'язку з цим є актуальною задача підвищення ефективності кодування широких класів зображень за рахунок використання нероздільних алгоритмів [5].

## 1.2 Представлення цифрових зображень

Комп'ютерне зображення в його цифровому представленні є набором значень інтенсивностей світлового потоку, розподілених по кінцевій площі.

Для простоти розглянемо спочатку монохромні зображення. Інтенсивність випромінювання світлової енергії з одиниці поверхні в точці з координатами  $(\xi, \eta)$  зображення можна представити деяким числом  $B(\xi, \eta)$ . Одиничний елемент зображення, що характеризується певним значенням  $(\xi, \eta)$ , називається пікселем, а величина  $z = f(\xi, \eta)$  – яскравістю [7].

Перш ніж розглянути алгоритми стиснення зображень, необхідно визначити що надалі буде розумітися під зображенням.

З математичної точки зору, зображення в градаціях сірого можна представити як дійсну функцію  $I$  двох дійсних змінних  $x$  і  $y$ . Функція  $I(x, y)$  зображення в загальному випадку визначається в прямокутній області, але для зручності досліджень в роботі всі зображення визначаються в квадратних областях, тобто  $x \in [0; W]$ , а  $y \in [0; H]$ , де  $W$  – ширина зображення, а  $H$  – висота зображення і  $W = H$ .

Всі зображення можна поділити на дві групи: з палітрою і без неї. У зображень з палітрою у пікселі зберігається число – індекс в деякому

одновимірному векторі кольорів, званому палітрою. Палітри зазвичай бувають 8, 16 і 256 кольорів [7].

Зображення без палітри зазвичай бувають в певній системі представлення кольору або в градаціях сірого. В градаціях сірого значення кожного з пікселів визначається як яскравість точки. Найбільш часто зустрічаються зображення з 2-ма, 16-ма і 256-ю рівнями сірого.

Якщо зображення відображається в якійсь системі представлення кольору, то кожен її піксель є структурою, що описує компоненти кольору. Найбільш поширеною системою представлення кольору, що використовується в електронних та комп'ютерних системах, є система RGB. У цій системі колір визначається як комбінація червоного, зеленого та синього кольору. І на кожному з складових припадає по одному байту. Існують і інші системи представлення кольору, такі як CMYK, CIE, YUV і YCrCb [4].

Для того, щоб коректніше оцінити ступінь стиснення зображення та застосовності того чи іншого алгоритму стиснення до даного зображення вводиться поняття класу зображення.

Під класом цифрового зображення розуміється сукупність зображень, застосування до яких алгоритму стиснення дає якісно однаковий результат. Наприклад, для одного класу алгоритм стиснення дає чудовий коефіцієнт стиснення, а для іншого класу зображень навпаки, збільшує об'єм стиснутого файлу [11].

У процесі роботи із зображеннями додатки, які здійснюють обробку, висувають різні вимоги до алгоритмів стиснення зображень. Через специфіку додатків такі вимоги іноді можуть суперечити один одному. У загальному випадку можна виділити наступні вимоги до алгоритмів стиснення зображень:

- високий ступінь компресії;
- висока якість стисненого зображення (дана вимога суперечить виконанню попередньої вимоги, тому завжди доводиться шукати компроміс між ступенем стиснення і якістю відновленого зображення);

- висока швидкість компресії (дана вимога актуально для додатків, що займаються кодуванням зображень в реальному масштабі часу: цифрових фотоапаратів, відеокамер);
- висока швидкість декомпресії (дана вимога актуально майже для всіх додатків);
- можливість показати приблизне зображення, не чекаючи повної його завантаження (дана вимога актуально для мережових додатків і для додатків, що займаються передачею великих зображень);
- врахування специфіки зображення (дану вимога реалізують алгоритми стиснення, засновані на визначенні «області особливого призначення» (ROI - regionsofinterest)).

### **1.3 Класифікація методів стиснення. Основні характеристики**

Всі методи стиснення інформації засновані на простому припущенні, що набір даних завжди містить надлишкові елементи. Стиснення досягається за рахунок пошуку та кодування надлишкових елементів.

Потік даних про зображення має істотну кількість зайвої інформації, яка може бути усунена практично без помітних для ока спотворень. При цьому розрізняють два типи надмірності.

Статистична надлишковість пов'язана з кореляцією і передбачуваністю даних. Ця надмірність може бути усунена без втрати інформації, вихідні дані при цьому можуть бути повністю відновлені. Найбільш відомі методи ефективного кодування символів засновані на знанні частоти кожного символу присутнього в повідомленні. Знаючи ці частоти, будують таблицю кодів, що володіє наступними властивостями:

- різні коди можуть мати різну кількість біт;
- коди символів, які зустрічаються з більшою частотою, мають менше біт, ніж коди символів з меншою частотою;

- хоча коди мають різну бітову довжину, вони можуть бути відновлені єдиним чином, тобто коди будуються як префіксні.

Цими властивостями володіє відомий алгоритм Хаффмана [7].

Візуальна (суб'єктивна) надлишковість, яку можна усунути з частковою втратою даних, мало впливають на якість відтворення зображень; це – інформація, яку можна вилучити з зображення, не порушуючи візуальне сприйняття якості зображень.

Завдання стиснення зображення складається з двох основних частин: кодування і декодування. Якщо декодоване зображення завжди в точності відповідає кодованому зображенню, то такий алгоритм кодування-декодування називається алгоритмом стиснення без втрат. Якщо декодувати зображення відрізняється від кодованого, то подібний алгоритм називають алгоритмом стиснення з втратами. Загальна схема процесу стиснення зображення представлена на рисунку 1.1.

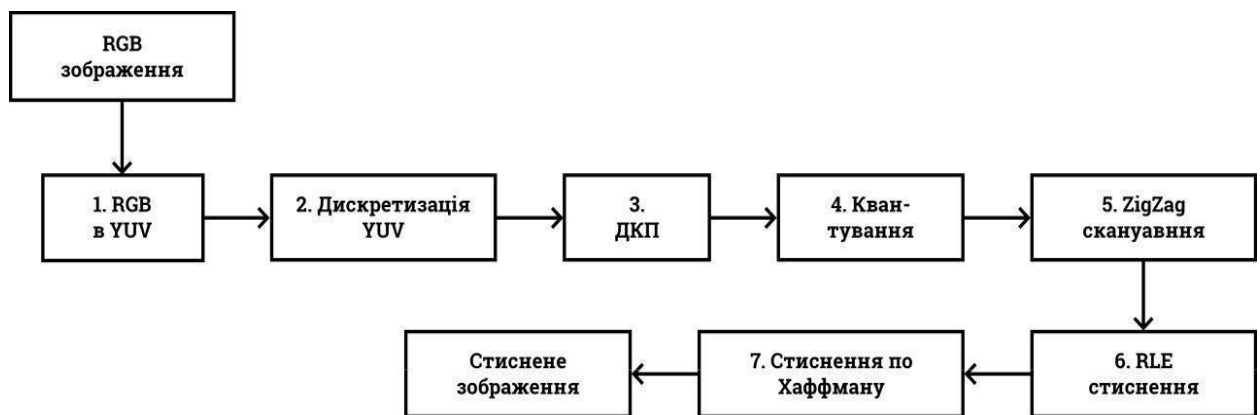


Рисунок 1.1 – Схема процесу стиснення зображень

Розглянемо етапи процедури стиснення даних в загальному вигляді. Будь-який метод стиснення реалізує три основних етапи:

- кодування або первинне стиснення;
- вторинне стиснення;
- декодування або відновлення зображення.



У додатку Б рисунок Б.1 наведено основні етапи стиснення цифрових зображень.

На першому етапі виконується перетворення вихідних даних з однієї форми подання в іншу. Зокрема, при стисненні зображень залежно від виду алгоритму стиснення може бути виконаний перехід від початкового зображення до наступних видів подання (додаток Б таблиця Б.1).

На другому етапі компоненти перетворення квантуються і зводяться до виду зручного для статистичного кодування, а потім кодуються. На цьому етапі забезпечується ущільнення інформаційного потоку.

Класифікація найбільш поширених методів стиснення наведена на рисунку 1.2.

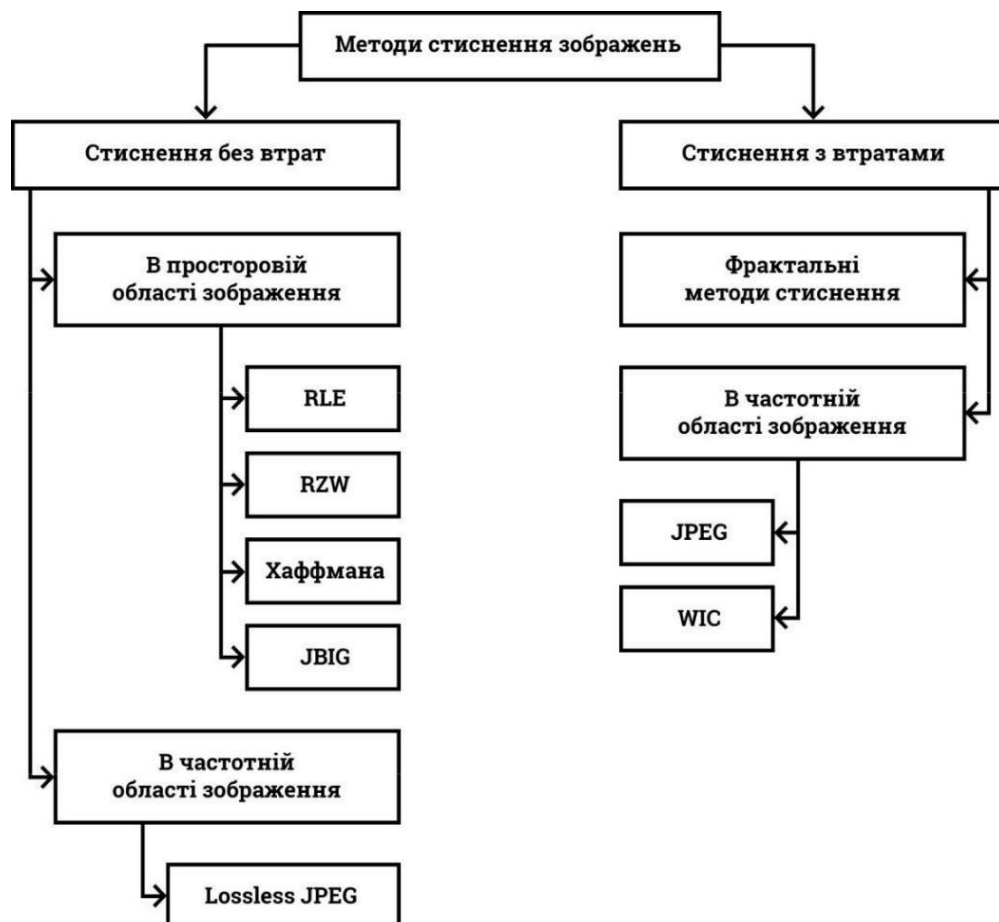


Рисунок 1.2 – Класифікація найбільш поширених методів стиснення

Результатом стиснення без втрат завжди є зниження обсягу вихідного потоку інформації без зміни його інформативності, тобто без втрати інформаційної структури. Більше того, з вихідного потоку, за допомогою відновлюючого алгоритму, можна отримати вхідний [7].

Під стисненням з втратами мається на увазі таке перетворення вхідного потоку даних, при якому вихідний потік, що базується на певному форматі інформації, представляє досить схожий за зовнішніми характеристиками на вхідний потік об'єкт, проте відрізняється від нього об'ємом. Ступінь подібності вхідного і вихідного потоків визначається ступенем відповідності деяких властивостей об'єктів, що представляється даним потоком інформації [10].

Для симетричних методів стиснення процедури стиснення і відновлення однотипні. Час стиснення і відновлення для таких методів порівнянні. Для несиметричних методів процедура стиснення відрізняється від процедури відновлення і зазвичай займає більше машинного часу.

#### **1.4 Алгоритми стиснення зображень без втрат**

Алгоритми стиснення без втрат - існують наступні алгоритми стиснення без втрат: алгоритм RLE, алгоритм LZW, алгоритм Хаффмана, JBIG, Lossless JPEG.

Коли говориться про стиснення без втрат, то мається на увазі, що існує алгоритм, зворотний алгоритму стиснення, що дозволяє точно відновити вихідне зображення.

##### **1.4.1 Алгоритм RLE**

Найбільш відомий і простий алгоритм стиснення інформації без втрат – це кодування серій послідовностей (Run Length Encoding – RLE) [9]. Суть даного

підходу полягає в заміні ланцюжків або серій повторюваних байтів або їх послідовностей на один кодуєчий байт і лічильник числа їх повторень. Проблема всіх аналогічних методів полягає лише у визначенні способу, за допомогою якого алгоритм розпакування міг би відрізнити в результуючому потоці байтів кодовану серію від інших – некодованих послідовностей байтів. Рішення проблеми досягається зазвичай постановкою міток на початку кодованих ланцюжків. Такими позначками можуть бути, наприклад, характерні значення бітів в першому байті кодованої серії, значення першого байта кодованої серії і т.п. Кращий, середній і найгірший коефіцієнти стиснення  $1/32$ ,  $1/2$ ,  $2/1$  [8].

Дані методи, як правило, досить ефективні для стиснення растрових графічних зображень (BMP, PCX, TIFF), так як останні містять досить довгі серії повторюваних послідовностей байтів.

Недоліком методу RLE є досить низька ступінь стиснення або вартість кодування файлів з малим числом серій  $i$ , що ще гірше— з малим числом повторюваних байтів в серіях. До позитивних сторін алгоритму, мабуть, можна віднести тільки те, що він не вимагає додаткової пам'яті при роботі, і швидко виконується. Цікава особливість групового кодування у форматі PCX полягає в тому, що ступінь архівації для деяких зображень може бути істотно підвищений всього лише за рахунок зміни порядку кольорів у палітрі зображень.

#### **1.4.2. Стиснення за методом Хаффмана**

Кодування Хаффмана частіше за інших вживається для текстових файлів, яке полягає в тому, що символи тексту замінюються ланцюжками біт різної довжини. Методика Хаффмана гарантує однозначну побудову коду з найменшим для даного розподілу ймовірностей середнім числом символів на букву [7].

Щодо стиснення зображень, то в основі такого методу лежить облік частоти появи однакових байт в зображенні. При цьому пікселям вихідного

зображення, які зустрічаються більше число разів, зіставляється код меншої довжини, а пікселям, що зустрічаються рідко – код більшої довжини (тобто формується префіксний код змінної довжини). Для збору статистики потрібно два проходи по файлу – один для перегляду та збору статистичної інформації, другий – для кодування. Коефіцієнти стиснення: 1/8, 2/3, 1 [11].

При використанні такого методу потребується запис у файл і таблиці відповідності кодованих пікселів та кодуєчих ланцюжків. Таке кодування застосовується в якості останнього етапу архівації в JPEG. Методи Хаффмана дають досить високу швидкість і помірно добру якість стиснення.

Основним недоліком даного методу є залежність ступеню стиснення від близькості ймовірностей символів до величини  $2^{-M}$ , оскільки кожен символ кодується цілим числом біт. Так, при кодуванні даних з двохсимвольним алфавітом стиснення завжди відсутнє, тому незважаючи на різні ймовірності появи символів у вхідному потоці алгоритм фактично зводить їх до 1/2. Такий алгоритм реалізований у форматі TIFF [3].

### **1.4.3. Алгоритм Лемпеля-Зива (LZ-compression)**

Суть даного алгоритму полягає в наступному: пакувальник постійно зберігає деяку кількість останніх оброблених символів у буфері. У міру обробки вхідного потоку новоприйняті символи потрапляють в кінець буфера, зсуваючи попередні символи і витісняючи найстаріші. Розміри цього буфера, званого також ковзаючим словником, варіюються в різних реалізаціях кодуєчих систем. Потім, після побудови хеш таблиць, виділяють (шляхом пошуку в словнику) найдовший початкову підрядок вхідного потоку, який збігається з одним з підрядків в словнику, і видають на вихід пару (length, distance), де length – довжина знайденого в словнику підрядка, а distance – відстань від нього до вхідного підрядка (тобто фактично індекс підрядка в буфері, вирахований з його

розміру). Якщо такий підрядок не знайдено, у вихідний потік просто копіюється черговий символ вхідного потоку [9].

Існує досить велике сімейство LZ-подібних алгоритмів, які розрізняються, наприклад, методом пошуку повторюваних ланцюжків. Один з досить простих варіантів цього алгоритму, наприклад, припускає, що у вхідному потоці йде або пара <лічильник, зсув щодо поточної позиції>, або просто <лічильник> байт “що пропускаються” і самі значення байтів [6].

При розархівації для пари лічильник, зміщення> копіюються <лічильник> байт з вихідного масиву, отриманого в результаті розархівації, на <зміщення> байт раніше, а <лічильник> (тобто число рівне лічильнику) значень байт “що пропускаються” просто копіюються у вихідний масив з вхідного потоку.

Даний алгоритм є несиметричним по часу, оскільки вимагає повного перебору буфера при пошуку однакових підрядків.

До переваг LZ можна віднести надзвичайну простоту алгоритму декомпресії.

#### **1.4.4. Алгоритм Лемпеля-Зива-Велч (Lempel-Ziv-Welch -LZW)**

Даний алгоритм відрізняється високою швидкістю роботи як при упаковці, так і при розпакуванні, досить скромними вимогами до пам'яті і простою апаратною реалізацією. Недолік – низький ступінь стиснення у порівнянні зі схемою двоступеневого кодування. Алгоритм перетворює потік символів на вхіді в потік індексів комірок словника на виході. Існує досить велике сімейство LZW-подібних алгоритмів, що розрізняються, наприклад, методом пошуку повторюваних ланцюжків [9].

Коефіцієнти стиснення: 1/1000, 1/4, 7/5. Коефіцієнт 1/1000 досягається тільки на одноколірних зображеннях розміром більше 4 Мб. Ситуація, коли алгоритм збільшує зображення, зустрічається вкрай рідко. Стиснення забезпечується за рахунок однакових підланцюжків у потоці. Алгоритм є майже

симетричним, за умови оптимальної реалізації операції пошуку рядка в таблиці. LZW універсальний – саме його варіанти використовуються в звичайних архіваторах. Він реалізований у форматах GIF, TIFF і TGA.

## **1.5 Алгоритми стиснення з втратами**

Алгоритми стиснення з втратами - до алгоритмів з втратами відносяться наступні алгоритми: алгоритм JPEG, фрактальний алгоритм, вейвлетний (рекурсивний або хвильовий) алгоритм. Для алгоритмів стиснення з втратами зворотного алгоритму не існує. Існує алгоритм, який відновлює зображення, яке не обов'язково точно збігається з вихідним.

### **1.5.1 Стиснення за стандартом JPEG**

JPEG – один з найпоширеніших і досить потужних алгоритмів, являє собою метод стиснення зображень, що реалізується різними способами [7,9]. Працює він як на чорно-білих, так і на повнокольорових зображеннях.

Стандарт JPEG (Joint Photographic Experts Group - об'єднана експертна група по фотографії) – формат зберігання фотографічних зображень, що відрізняється гарною якістю відновленого зображення. Процес стиснення зображення JPEG досить складний і часто для досягнення прийнятної продуктивності вимагає спеціальної апаратури. Схема процедури стиснення зображень за стандартом JPEG наведена у додатку В.

Кодування зображень за стандартом JPEG зазвичай починається з перетворення колірного простору з RGB в YUV (відоме також під назвою YCbCr). Кольорове зображення традиційно може розглядатися як результат складання трьох компонент:

$$X_{ijC} = a_1 X_{ij(R)} + a_2 X_{ij(G)} + a_3 X_{ij(B)} \quad (1.1)$$

У цьому виразі  $a_1, a_2, a_3$  — калориметричні коефіцієнти.

У типових зображеннях у форматі RGB є істотна кореляція між кольоровими компонентами і з точки зору стиснення зображення цей формат є завідомо надлишковим. Як відомо, в стандартах телевізійного мовлення використовується інше представлення зображень, при якому також використовуються 3 компоненти сигналу, але при цьому ці компоненти майже некорельовані один з одним. Компоненти  $R, G$  і  $B$  перетворюються в яскравісну компоненту  $Y$  і дві різнокольорові компоненти  $U$  і  $V$ , формату YUV.

### 1.5.2 Фрактальне стиснення зображень

Основу фрактального підходу становить постулат про те, що зображення реального світу мають афінну надмірність [7]. Інакше кажучи, існує набір афінних коефіцієнтів, що описують обертання, стиснення, розширення, спотворення форми, зрушення об'єктів зображення.

Найбільш поширеним прикладом фрактальної зображення, згенерованого за допомогою IFS є зображення папороті (рис.1.3), використане для створення даного зображення, складається з 4-х афінних перетворень. Кожне перетворення кодується ліченими байтами, хоча вихідне зображення може бути будь-якого розміру. Таким чином, можна зробити висновок, що фрактальна компресія – це пошук самоподібних областей та визначення для них параметрів афінних перетворень [4].

Проста IFS, що породила зображення папороті не придатна для стиснення довільних зображень, оскільки вони зазвичай подаються у градаціях сірого, а не з значень 0 або 1 і тому прості IFS використовуються тільки для самоподібних зображень, тобто коли зображення будуються з елементів, що є копією цілого зображення.



Рисунок 1.3 – Зображення папороті, що згенеровано за допомогою IFS

В основі фрактального стиснення лежать кілька основних визначень і теорем. Розглянемо їх коротко.

Перетворення. Перетворення зіставляє точки в одному просторі точку в іншому просторі, можливо в тому ж самому просторі. Перетворення називається відображенням  $f$  і записується як  $f: X_1 \rightarrow X_2$  якщо воно переводить простір  $X_1$  в простір  $X_2$ .

Перетворення в метричному просторі  $(X, d)$  називається стискующим відображенням, якщо існує константа  $s$ ,  $0 \leq s < 1$  така що:

$$d(f(x_1), f(x_2)) \leq s \cdot d(x_1, x_2) \quad (1.2)$$

де  $d(x_1, x_2)$  - відстань від точки  $x_1$  до точки  $x_2$  в просторі  $X$ .

Константа  $s$  називається коефіцієнтом стиснення відображення  $f$ .

На рисунку 1.4 показаний приклад стиснення відображення  $(R_2, d_2)$ , застосованого до безлічі точок в  $R_2$ . Тут дане перетворення застосовувалося більше одного разу: спочатку  $f(x)$  обчислювалося для точки  $x$ , а потім перетворення  $f$  застосовувалося до результату перетворення:  $f(f(x))$ . Таке послідовне, багаторазове застосування перетворення називають ітераціями і позначають як:  $f^n$ , тобто  $f(\dots f(x) \dots)$ , де  $f$  застосовується  $n$  раз.



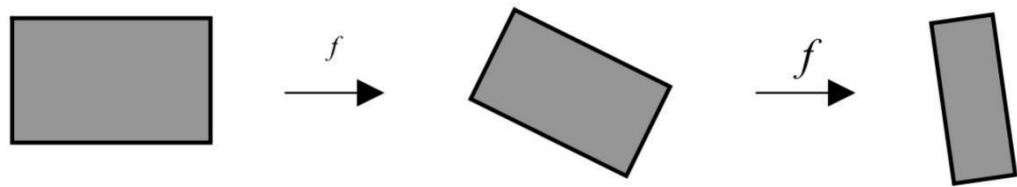


Рисунок 1.4 – Стискаюче відображення для точок у просторі  $R^2$

Система ітеруючих функцій (IFS) складається з повного метричного простору  $(X, d)$  і кінцевого безлічі стискають відображень  $w_n: X_1 \rightarrow X_2$  з коефіцієнтами стиснення  $s_n$ . Таким чином, IFS можна позначити таким чином:  $\{X, w_n: n = 1, 2, \dots, N\}$ , або якщо розглядати простір точок в  $R^2$ , то просто  $\{w_n\}$ .

У теорії фрактального стиснення доводиться теорема колажу, яку можна сформулювати наступним чином [14].

Нехай  $\epsilon$  двійкове зображення і нехай  $\epsilon$  стискають відображення, такі що:

$$U_n^N = \bigcap_{n=1}^N w_n(L)$$

Покривають  $L$  майже точно. Можна вважати таке  $w_n(L)$  зменшеною копією  $L$ . Тоді теорема колажу стверджує, що аттрактор  $A$  (аттрактором IFS називається зображення, яке є єдиною нерухомою точкою IFS) системи  $\{w_n\}$  близький до  $L$ . «Колаж» є набір областей  $w_n(L)$ .

Так як аттрактор  $A$  – це результат нескінченного числа ітерацій IFS, то він є за своєю суттю фракталом.

Щоб на практиці застосувати теорему колажу (рис.1.4), необхідно вибрати перетворення, які будуть стискаючими відображеннями. Одним з таких перетворень є афінне перетворення:  $T: R^2 \rightarrow R^2$ :

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \check{x} \\ \check{y} \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (1.3)$$

де  $a, b, c, d, e, f \in \mathbb{R}$ . Афінні перетворення можуть здійснювати поворот, переміщення й масштабування (рисунок 1.5).

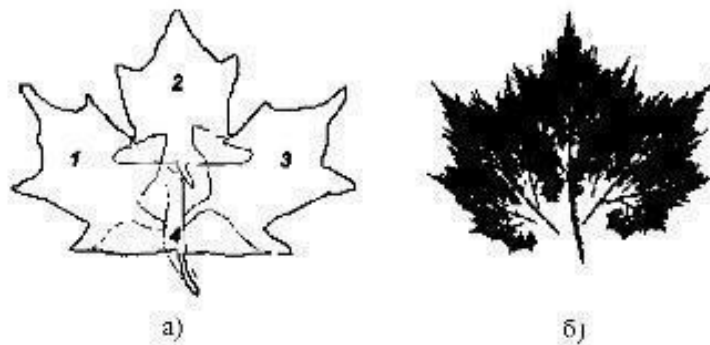


Рисунок 1.5 – Ілюстрація теореми колажу. (а) вихідне зображення і чотири фрагменти зображення; (б) зображення-аттрактор

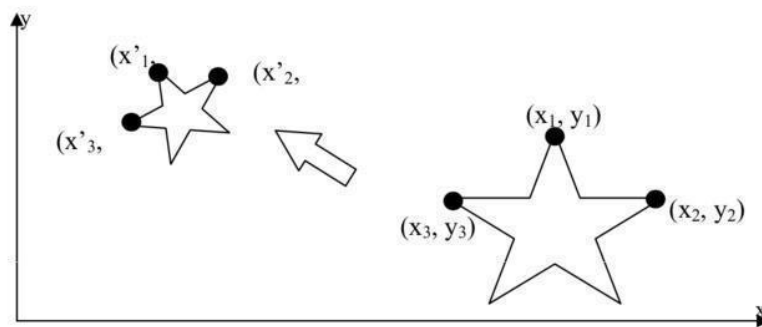


Рисунок 1.6 – Дія афінного перетворення на точки в просторі  $R^2$

Щоб створити IFS зображення – потрібно знайти хоча б деяку приблизну самоподібність в зображенні. Потім потрібно виділити точки і перетворення IFS для кожної з фігур подібності на зображенні. Коли точки і перетворення для IFS визначені, то обчислюються коефіцієнти афінного перетворення, використовуючи систему рівнянь, засновану на виразі (1.4).

Існує два алгоритми побудови зображення-аттрактора за допомогою IFS. Один з них це пряме застосування теореми про стискають відображеннях, а інший - застосування так званої «гри хаосу». Детермінований алгоритм для побудови зображення є аттрактором IFS, до будь-якого початкового зображенню

В застосовує теорему про стиснення відображень. Алгоритм будує послідовність зображень  $A_n$ , багаторазово застосовуючи IFS відображення  $W = \{w_1, w_2, \dots, w_n\}$ :

$$A_n = W^{on}(B) \quad (1.4)$$

Детермінований алгоритм корисний з точки зору навчання, оскільки дозволяє бачити результат перетворення на кожному кроці ітерації (рисунок 1.6). Імовірнісний алгоритм пов'язує з кожним афінним перетворенням  $w_i$  в IFS ймовірність  $p_i$ . Ці ймовірності визначають, наскільки щільно кожна частина зображення-атрактора покрита точками [15]. Імовірнісний алгоритм створює зображення-атрактори високої якості швидше, ніж детермінований алгоритм. Це відбувається не тільки через те, що імовірнісний алгоритм на кожному кроці ітерації виконує меншу роботу, а й сама виконана робота дає кращий результат.

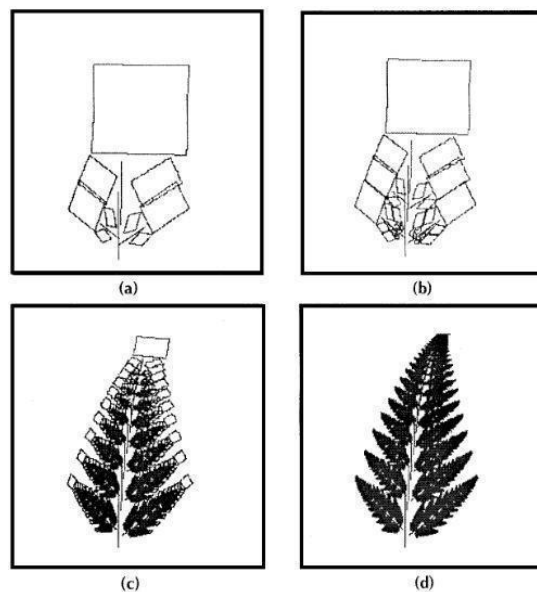


Рисунок 1.7 – Детермінований алгоритм, застосований до IFS папороті.

Вид зображення  $A_n$  після: а) 2-х, б) 3-х, в) 10-ї, г) 30-ї ітерацій.

Реальні зображення не володіють властивістю глобальної самоподібності, яка притаманна зображенням IFS. До того ж реальні зображення представлені в градаціях сірого, а не в двійкових значеннях, як зображення IFS. Тому при

фрактальному кодуванні реальних зображень намагаються знайти безліч стискають перетворень, які відображають доменні блоки (які можуть перекриватися) в безліч рангових блоків, які покривають зображення. Рангові блоки можуть бути однакового розміру, але частіше використовується адаптивне розбиття з перемінним розміром блоку (одним з поширених методів адаптивного розбиття є метод квадродерева) [16].

Базовий алгоритм фрактального кодування зображення виконується таким чином (рисунок 1.8):



Рисунок 1.8 – Відображення доменних блоків у рангові при фрактальному стисненні зображення

- Зображення  $f$  розбивається на не перекриваються рангові блоки  $\{R_i\}$ . У самому простому випадку блоки можуть являти собою прямокутники, але можуть бути й інші форми.
- Зображення покривається послідовністю доменних блоків, які можуть перетинатися. Домени можуть бути різного розміру, і їх кількість може обчислюватися сотнями і тисячами.
- Для кожного рангового блоку знаходять домен і відповідне перетворення, яке найкращим чином покриває ранговий блок. Зазвичай це Афінний перетворення.
- Якщо достатньо точної відповідності не вийшло, то розбиваємо рангові блоки на менші рангові блоки. Даний процес продовжується до тих пір поки, не

отримують прийнятної відповідності або розмір рангового блоку досягає певних значень.

В даний час відомі наступні різновиди такого алгоритму [9].

Алгоритм Фішера. Ідея алгоритму полягає в тому, щоб певним чином класифікувати  $D$ -блоки і  $R$ -блоки, а пошук близького  $D$ -блоку виробляти в тому ж класі, до якого відноситься рангова область. Робиться це в такий спосіб [9].

Вихідні блоки розбиваються на чотири частини. Для кожної з частин підраховується середнє значення  $A_i$  і дисперсія  $V_i$  пікселів. Далі блоки класифікуються за наступним принципом.

Визначимо три базових типи блоків:

$$\begin{aligned} \text{тип 1: } & A_1 \geq A_2 \geq A_3 \geq A_4, \\ \text{тип 2: } & A_1 \geq A_2 \geq A_4 \geq A_3, \\ \text{тип 3: } & A_1 \geq A_4 \geq A_2 \geq A_3. \end{aligned} \quad (1.5)$$

Зрозуміло, що будь-який блок за допомогою відповідного афінного перетворення квадрата в квадрат можна привести до виду, відповідному одному з вказаних типів. Після того, як ми зафіксували три основні класи, блоки класифікуються за дисперсії. Таким чином, в кожному з трьох класів з'являються 24 підкласу, разом 72 класу. Пошук близького до  $R$ -блоку  $D$ -блоку проводиться перебором у відповідному класі [17].

Популяцією  $\Pi = (\chi^1, \chi^2, \dots, \chi^M)$  чисельності  $M$  вважається вектор простору  $B^M$ , координати якого називаються генотипами особин даної популяції.

Кроком ГА є перехід від поточного покоління до наступного, тобто отримання нової популяції  $\Pi_{t+1}$  з  $\Pi_t$ . У побудові чергової особини нової популяції беруть участь оператори кросинговеру (схрещування), мутації і випадковий оператор відбору,  $Select : B^M \rightarrow \{1, \dots, M\}$  дія якого полягає у виборі номера особини батька при породженні чергового нащадка.

Для визначення ГА необхідно задати оператор кросинговеру  $Cross : B \times B \rightarrow B \times B$  і оператор мутації  $Mut : B \rightarrow B$ .

Дія кросинговеру  $(\chi', \tau') = \text{Cross}(\chi, \tau)$  полягає у виборі випадковим чином деякої позиції  $j$ , рівномірно розподіленої від 1 до  $N-1$ , після чого результат формується у вигляді

$$\chi' = (\chi_1, \chi_2, \dots, \chi_j, \tau_{j+1}, \dots, \tau_N), \tau' = (\tau_1, \tau_2, \dots, \tau_j, \chi_{j+1}, \dots, \chi_N). \quad (1.6)$$

Вплив кросинговеру регулюють за допомогою ймовірності  $P_{\text{Cross}}$  спрацьовування цього оператора (в іншому випадку все залишається без змін).

Оператор мутації в кожній позиції аргументу із заданою вірогідністю  $P_{\text{mut}}$  замінює її вміст на випадковий елемент двійкового алфавіту  $A$ , обраний відповідно з рівномірним розподілом (в іншому випадку все залишається без змін).

Цільова функція вихідної задачі, замінюється в ГА на функцію придатності генотипу  $\Phi(\chi)$ , де  $\chi \in B$ .

Процес роботи алгоритму є послідовна зміна поколінь, на кожному кроці якої популяція  $\Pi_{t+1}$  наповнюється парами нащадків від особин популяції  $\Pi_t$  за формулою:

$$(\chi^{t+1}_k, \chi^{t+1}_{k+1}) = \text{Mut}(\text{Cross}(\chi^{t \text{ Select}(\Pi^t)}, \chi^{t \text{ Select}(\Pi^t)})) \quad (1.7)$$

де  $(\chi^{t+1}_k, \chi^{t+1}_{k+1})$  – особини з найменшою придатністю популяції  $\Pi$ . Тобто індивіди витягуються попарно з  $\Pi^t$  і після кросинговеру і мутації поміщаються в  $\Pi^{t+1}$ . Зміна ймовірностей мутації і кросинговеру дозволяє регулювати роботу ГА і налаштовувати його на конкретні завдання [18].

Збільшення швидкодії алгоритму. Недоліком даного методу стиснення є дуже великий обсяг обчислень для кодування зображень. Для програм реалізують даний метод, застосовуються як звичайні методи збільшення швидкості програм: цілочисельна арифметика, асемблерна реалізація та інші, так і деякі специфічні прийоми: використання афінних перетворень тільки одного типу, різні варіанти розбиття зображення на області  $R_i$ .

### 1.5.3 Метод усіченого блочного кодування

Назва методу відображає той факт, що зображення розбивається на невеликі прямокутні ділянки однакового розміру, які називають блоками. Цей метод на відміну від більшості інших підлаштовує параметри кодування не під деяку усереднену характеристику всього зображення, а під локальні особливості в межах кожного блоку. Це дозволяє зберегти дрібні деталі зображень. Метод не призводить до розмивання меж, що є характерним для деяких інших алгоритмів. Метод УБК порівняний з більшістю інших методів по ефективності стиснення даних і за обсягом обчислень, необхідних для кодування, але не має конкурентів за простотою декодування.

Базовий алгоритм УБК будується наступним чином [9]. Зображення, представлене  $M \times N$  – матрицею  $\|b_{ij}\|$  яскравості пікселів, розбивається на невеликі прямокутні блоки  $m \times n$  елементів. Кожен такий блок обробляється незалежно від інших, тому опишемо алгоритм обробки одного блоку.

Обробка блоку починається з обчислення порога і двох рівнів квантування, потім проводиться квантування блоку на два рівня, після чого слідує упаковка проквантованого блоку. Для визначення рівнів квантування спочатку обчислюються два перших вибіркового моменти – середнє значення  $C$  і середній квадрат  $E$ :

$$C = \frac{1}{m \times n} \sum_i \sum_j b_{ij}, \quad E = \frac{1}{m \times n} \sum_i \sum_j b_{ij}^2 \quad (1.9)$$

(де підсумовуються елементи зображення в межах блоку) і дисперсія

$$\sigma^2 = E - c^2 \quad (1.10)$$

Порогова величина квантування  $d$  покладається рівною середньому  $C$ . Верхній  $a$  і нижній  $b$  рівні квантування обчислюються за такими формулами:

$$a = C - \sigma\sqrt{q/(p - q)}, \quad b = C + \sigma\sqrt{(p - q)/q} \quad (1.11)$$

де  $p = m \times n$  – число елементів блоку,  $q$  – число елементів, що перевищують поріг  $d$ .

Квантування проводиться за правилом:

$$s_{ij} = \begin{cases} a, & \text{якщо } b_{ij} < d \\ b, & \text{якщо } b_{ij} \geq d \end{cases} \quad (1.12)$$

де  $s_{ij}$  – елементи зображення після квантування.

Після квантування виходить блок, що містить тільки рівні  $a$  та  $b$ . Незавжно показати, що середнє значення і середній квадрат вихідного і проквантованого блоків збігаються. Практично для зручності подальшої упаковки замість  $a$  записується нуль, замість  $b$  – одиниця. Рівні  $a$  і  $b$  записуються окремо [19].

Упаковка полягає в тому, що блок, який містить лише нулі і одиниці, інтерпретується як двійкове число, що має розрядів. Відновлення закодованого зображення також проводиться по блоках і полягає в розпакуванні і зворотній підстановці.

З алгоритму видно, що ступінь стиснення безпосередньо залежить від розмірів блоку. Найбільш задовільні результати, як за ступенем стиснення, так і за якістю відновленого зображення були отримані при використанні блоків розміром  $4 \times 4$  [11].

Описаний вище спосіб визначення порогу і рівнів квантування НЕ є єдиним. Існує ряд інших критеріїв. Важливо, щоб критерій відповідав цілям подальшої обробки зображення і її конкретним особливостям.



## 1.6 Вибір і обґрунтування аналогу

В якості аналога до розробленого алгоритму стиснення зображень на основі дискретних ортогональних функцій обрано метод стиснення зображень JPEG. JPEG – є методом стиснення, що дозволяє стискати дані багатоградацийних зображень з піксельною глибиною від 6 до 24 біт з задовільною швидкістю й ефективністю. Схема JPEG заснована на відкиданні інформації, яку важко помітити візуально. Міра стиснення зображення з фотографічною якістю може становити від 20:1 до 25:1 без помітної втрати якості. Звичайно ж, настільки високий показник стиснення супроводжується відмінністю від оригіналу, але вона настільки незначна, що якість зображення все-таки залишається досить високою. Зображення, що містять великі області одного кольору, стискаються дуже погано.

Недоліками аналогу є:

- при підвищенні ступені стиснення зображення розпадається на окремі квадрати (8 x 8, 16 x 16). Це пов'язано з тим, що відбуваються великі втрати в низьких частотах при квантуванні, і відновити вихідні дані стає неможливо.
- проявляється «ефект Гіббса» – ореоли на границях різких переходів кольорів [17].

## 1.7 Вибір напрямку та задач дослідження

Методи стиснення зображення поділяються на дві великі групи – методи стиснення без втрат та методи стиснення з втратами.

Методи стиснення без втрат дозволяють отримати відновлене зображення ідентичне вихідному. Такі методи стиснення використовують тоді, коли ідентичність вхідного та вихідного зображень є критичною або коли самі шуми зображення є об'єктом дослідження – наприклад, в медицині та науковій

діяльності. Методи стиснення без втрат дають в результаті невеликі коефіцієнти стиснення [20].

Методи стиснення з втратами використовують у випадках, коли зміни зображення, які непомітні людському оку є прийнятними фото- та відеокамери, особливо веб. Такі зміни відновленого зображення дають змогу досягти значно більших коефіцієнтів стиснення, ніж при стисненні без втрат.

Проаналізовано сучасні методи стиснення зображень та виконано вибір аналогу.

Метою даної роботи є підвищення коефіцієнту стиснення цифрових зображень шляхом модифікації алгоритму стиснення JPEG, що дозволяє ефективно стискати зображення при мінімальному спотворенні.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- Провести аналіз існуючих методів стиснення зображень.
- Розробити модифікований метод стиснення на основі алгоритму JPEG.
- Провести експериментальне дослідження розробленого методу.
- Розробити алгоритмічне та програмне забезпечення.

## РОЗДІЛ 2

### ЗАСТОСУВАННЯ ОРТОГОНАЛЬНИХ ПЕРЕТВОРЕНЬ ДЛЯ СТИСНЕННЯ ЗОБРАЖЕНЬ

У загальному вигляді процедура лінійного перетворення вектору  $X = [x_1, x_2, \dots, x_n]^T$  запишеться в такий спосіб:

$$Y = HX, \quad (2.1)$$

де  $H$  – матриця перетворення розміром  $N \times N$  елементів.

Для того, щоб перетворення (1.4) було зворотнім, тобто  $X = H^{-1}Y$ , необхідно і достатньо існування матриці  $H^{-1}$ . Серед операторів перетворення особливе місце займають оператори ортогонального перетворення, для яких справедливе умова  $H^{-1} = H^T$ , де  $T$  – знак транспонування.

У задачах кодування зображень вигляд оператора перетворення визначає якість відновленого зображення при заданих коефіцієнтах стиснення. Тому необхідно підібрати таке перетворення, щоб відновлене зображення якнайменше відрізнялося від вихідного відповідно до обраної міри втрат [12].

#### 2.1 Перетворення Адамара

Простим методом перетворення вихідного сигналу є перетворення Адамара. Матриця перетворення  $H$  для двох випадкових величин буде мати вигляд:

$$H = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \quad (2.2)$$

Якщо з елементів матриці  $H$  скласти базисні вектори  $e_1 = [h_{11}, h_{12}] = [1, 1]$ ,  $e_2 = [h_{21}, h_{22}] = [1, -1]$ , то вони будуть характеризувати поворот ортогональної системи координат на  $45^\circ$  відносно одиничного базису (рисунок 2.1).

Якщо випадкові величини  $x_1$  та  $x_2$  мають кореляційну залежність, то проекція вектору  $X = [x_1, x_2]^T$  на базисний вектор  $e_1$  буде, в середньому, більше, ніж проекція цього ж вектору на базисний вектор  $e_2$ . Завдяки цьому інформація буде зосереджувати в першому коефіцієнті перетворення. Другий коефіцієнт слугує для уточнення представлення вектору  $X$  у новому базисі [21].

При збільшенні розмірності простору більша частина інформації буде зосереджена в малому числі коефіцієнтів, які потрібно зберегти з найменшими втратами. Інші коефіцієнти можна або відкинути, або квантувати із більшим кроком.

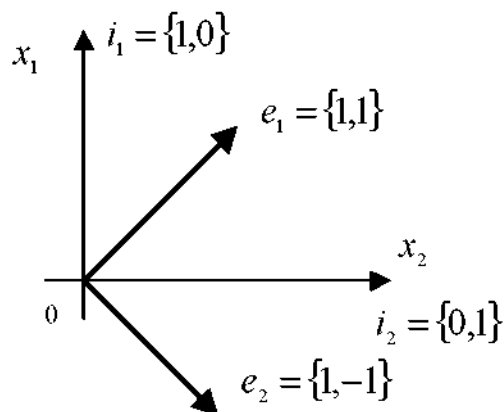


Рисунок 2.1 – Розташування базисних векторів перетворення Адамара щодо одиничного базису

Матрицю Адамара розмірності  $4 \times 4$  елементів легко побудувати з матриці Адамара  $H$  розмірністю  $2 \times 2$  елементи:

$$H_4 = \begin{vmatrix} H & H \\ H & -H \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \quad (2.3)$$

Користуючись співвідношенням (2.3), можна побудувати матрицю Адамара будь-якої розмірності  $N = 2^n$ , де  $n$  – будь-яке додатне число [22].

При аналізі зображень за допомогою даного перетворення спочатку обчислюються коефіцієнти по рядках зображення, а потім по стовпцях. У результаті виходить роздільне перетворення виду:

$$Y = HAN^T, \quad (2.4)$$

де  $A$  –  $N \times N$  матриця вихідного зображення.

Так як матриця  $H$  являє собою оператор ортогонального перетворення, то зворотне перетворення з  $Y$  в  $A$  запишеться у вигляді:

$$A = H^T Y H \quad (2.5)$$

Перевагою такого перетворення є простота реалізації й низька обчислювальна складність. Розглянуте розкладання буде давати гарні результати для кусково-постійних функцій, тому що базисні функції перетворення Адамара виділяють постійні складові сигналу. У результаті для кусково-постійних сигналів більша частина коефіцієнтів розкладання перетворення Адамара буде близька до нуля, що приведе до зменшення ентропії перетворених даних і збільшенню коефіцієнта стиснення. Однак на практиці такі сигнали зустрічаються досить рідко, тому виникає необхідність визначити таке перетворення, що давало б оптимальне або близьке до оптимального розкладання сигналів, що більш точно характеризує реальні зображення. При цьому під оптимальністю розуміється мінімум середньої дисперсії помилки

відновлення, при відкиданні частини високочастотних коефіцієнтів розкладання [13].

## 2.2 Перетворення Карунена-Лоева

У випадку ортогонального перетворення з набором  $N$  базисних векторів  $\phi_i, i = \overline{1, N}$  перетворення з  $Y$  в  $X$  можна записати як

$$X = A^T Y = \sum_{i=1}^N \phi_i y_i, \quad (2.6)$$

де  $A - N \times N$  - матриця ортогонального перетворення.

З усієї множини значень  $Y = [y_1, y_2, \dots, y_N]^T$  запам'ятовуються перші  $M$ , причому  $M < N$ . Після відновлення отримаємо оцінку вектору  $\hat{X}$ :

$$\hat{X} = \sum_{i=1}^M y_i \phi_i + \sum_{i=M+1}^N b_i \phi_i, \quad (2.7)$$

де  $b_i$  - деяке константне значення відкинутих коефіцієнтів розкладання  $y_i, i = \overline{M+1, N}$ .

Помилка при відкиданні  $N - M$  коефіцієнтів визначається як

$$\varepsilon = \Delta X = X - \sum_{i=1}^M y_i \phi_i - \sum_{i=M+1}^N b_i \phi_i = \sum_{i=M+1}^N (y_i - b_i) \phi_i, \quad (2.8)$$

а дисперсія цієї помилки

$$M\{\varepsilon^2\} = M\left\{\left(\sum_{i=M+1}^N (y_i - b_i)\phi_i\right)^2\right\} = M\left\{\sum_{i=M+1}^N \sum_{j=M+1}^N (y_i - b_i)(y_j - b_j)^T \phi_i \phi_j\right\} \quad (2.9)$$

Аналіз виразу (2.9) показує, що для мінімізації дисперсії помилки необхідно підібрати відповідні значення  $b_i$  та вектори  $\phi_i$ .

Для знаходження оптимальних векторів  $\phi_i, i = \overline{1, N}$ , перепишемо вираз (2.9) з урахуванням  $b_i = \phi_i^T M\{X\} = \phi_i^T \overline{X}$  та  $\phi_i^T \phi_j = 0$  при  $i \neq j$ ; у вигляді

$$\begin{aligned} M\{\varepsilon^2\} &= \sum_{i=M+1}^N M\{(y_i - b_i)(y_i - b_i)^T\} = \sum_{i=M+1}^N M\{(\phi_i^T X - \phi_i^T \overline{X})(\phi_i^T X - \phi_i^T \overline{X})^T\} = \\ &= \sum_{i=M+1}^N \phi_i^T M\{(X - \overline{X})(X - \overline{X})^T\} \phi_i = \sum_{i=M+1}^N \phi_i^T R_{xx} \phi_i \end{aligned} \quad (2.10)$$

де  $R_{xx} = M\{(X - \overline{X})(X - \overline{X})^T\}$  представляє собою коваріаційну матрицю. Для мінімізації (2.10) застосовується метод множників Лагранжа. В результаті отримаємо

$$R_{xx} \phi_i = \beta_i \phi_i \quad (2.11)$$

Вираз (2.11) визначає вектори  $\phi_i$  та значення  $\beta_i$  як власні вектори і власні значення кореляційної матриці  $R_{xx}$ . Таким чином матриця перетворення  $A$ , складена з власних векторів кореляційної матриці  $R_{xx}$ , дозволяє виконати перетворення Карунена-Лоева (ПКЛ) [14], яке мінімізує середню дисперсію помилки при відновленні сигналу по неповних даних.

Недоліком описаного методу є необхідність апріорних відомостей про кореляційну матрицю  $R_{xx}$  вихідного сигналу  $X$ , а також відносно висока обчислювальна складність. Так, у випадку застосування роздільного ПКЛ, для обчислення всіх коефіцієнтів перетворення потрібно  $N^2$  арифметичних операцій, де  $N$  кількість відліків вихідного сигналу  $X$ .

Таким чином, розглянутий метод може бути застосований у задачах, у яких апіорі відома кореляційна матриця  $R_{xx}$ , а також немає великих обмежень на обчислювальні ресурси.

### 2.3 Дискретне косинусне перетворення

Розглянуті способи перетворення зображень не повністю задовольняють вимогам що звичайно пред'являють до алгоритмів стиснення, таким як низька обчислювальна складність, високі коефіцієнти стиснення, гарна візуальна якість відновленого зображення. Ці вимоги обумовлюють вибір перетворень, які, з одного боку, мають низьку обчислювальну складність, а з іншого, добре описують сигнал, який кодується. Традиційно в якості одного з таких перетворень застосовують розкладання сигналу по косинусним гармонікам ряду Фур'є [15]. Даний метод одержав широке поширення й використовується в алгоритмі стиснення зображень JPEG.

Відомо, що ряд Фур'є для будь-якої безперервної дійсної й симетричної (парної) функції містить тільки дійсні коефіцієнти, що відповідають косинусним членам ряду. У відповідній інтерпретації цей результат можна поширити й на перетворення зображень. Існує простий спосіб одержання симетричних зображень, при якому до вихідного зображення впритул пристроюють його дзеркальні відбиття. При цьому з первісного масиву, що містить  $N \times N$  елементів, виходить масив з  $2N \times 2N$  елементів [23].

Елементи симетричного масиву пов'язані з елементами вихідного масиву співвідношеннями:

$$f_s(j, k) = \begin{cases} f(j, k) \text{ при } j \geq 0, k \geq 0, \\ f(-1-j, k) \text{ при } j < 0, k \geq 0, \\ f(j, -1-k) \text{ при } j \geq 0, k < 0, \\ f(-1-j, -1-k) \text{ при } j < 0, k < 0. \end{cases} \quad (2.12)$$



Побудований таким чином масив  $f_s(j, k)$  симетричний відносно точки  $j = -1/2, k = -1/2$ . Перетворення Фур'є для випадку, коли початок координат знаходиться в центрі симетрії, запишеться як:

$$F_s(u, v) = \frac{1}{2N} \sum_{j=-N}^{N-1} \sum_{k=-N}^{N-1} f_s(j, k) \exp \left\{ -\frac{2\pi i}{2N} \left[ u \left( j + \frac{1}{2} \right) + v \left( k + \frac{1}{2} \right) \right] \right\} \quad (2.13)$$

де  $u, v = \overline{-N, N-1}$ .

Оскільки масив  $f_s(j, k)$  симетричний і складається з дійсних чисел, відношення (2.13) можна записати у вигляді

$$F(u, v) = \frac{2}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cos \left[ \frac{\pi}{N} u \left( j + \frac{1}{2} \right) \right] \cos \left[ \frac{\pi}{N} v \left( k + \frac{1}{2} \right) \right] \quad (2.14)$$

Так як базисна функція при  $u = 0, v = 0$  представляє собою постійну складову й у два рази більше стосовно всіх інших базисних функцій парного косинусного перетворення, то при виконанні зворотного перетворення необхідно ввести нормуючий коефіцієнт  $C(t)$ , який визначається виразом

$$C(t) = \begin{cases} 1/2, & t = 0 \\ 1, & t > 0 \text{ \& } t < 0 \end{cases} \quad (2.15)$$

В цьому випадку зворотне перетворення запишеться наступним чином

$$f(j, k) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos \left[ \frac{\pi}{N} u \left( j + \frac{1}{2} \right) \right] \cos \left[ \frac{\pi}{N} v \left( k + \frac{1}{2} \right) \right] \quad (2.16)$$

Перевагою ДКП є те, що базисні функції цього перетворення належать поліному Чебишева й можуть бути обчислені рекуррентно. Крім того, це дозволяє синтезувати алгоритми швидкого ортогонального перетворення [25].

Недоліком ДКП є нерівномірна збіжність на краях інтервалу розкладання  $N$  при відкиданні коефіцієнтів, що позначається на візуальній якості відновленого зображення у вигляді артефактів, тобто областей, які не були присутні раніше на вихідному зображенні.

Розглянутий метод одержав широке поширення завдяки наявності швидкого алгоритму обчислення коефіцієнтів. Крім того, він не вимагає апріорної інформації про вхідний сигнал, на відміну від ПКЛ. ДКП рекомендується застосовувати для вхідних сигналів, які можуть бути добре описані косинусними гармоніками, а також мають період рівний періоду розкладання  $N$  [16].

## 2.4 Дискретне вейвлет-перетворення

Дискретне вейвлет-перетворення (ДВП) дає сигналу багатошкільне представлення засноване на ідеї обережного виявлення границь у вейвлет-теорії. Його переваги – трансляційна інваріантність, може характеризувати локальну форму нерегулярних структур розвитком локальних максимумів вейвлетів для звичайних сигналів [24].

Нехай  $\Psi(x)$  – вейвлет функція та  $f(x) \in L^2(\mathbb{R})$ . Дилатацію  $\Psi(x)$  з показником  $2^j$  можна визначити як [24]:

$$\psi_{2^j}(x) = \frac{1}{2^j} \psi\left(\frac{x}{2^j}\right). \quad (2.17)$$

Вейвлет перетворення для  $f(x)$  у масштабі  $2^j$  та позиції  $x$  визначається згорткою [24]:

$$W_{2^j} f(u) = \left( f(x) * \psi_{2^j}(x) \right)(u) = \int_{-\infty}^{\infty} f(x) \frac{1}{\sqrt{2^j}} \psi\left(\frac{x-u}{2^j}\right) dx. \quad (2.18)$$

Це означає, що масштаб для неперервного вейвлет-перетворення дискретний вздовж діадної послідовності  $\{2^j\}$ , причому  $j$  належить цілочисленній множині  $Z$ .

Діадне вейвлет-перетворення – це послідовність функцій [24]:

$$Wf = \left( W_{2^j} f(x) \right)_{j \in Z}, \quad (2.19)$$

де  $W$  – оператор діадного вейвлет-перетворення.

Нехай функція  $\phi(x)$  відповідна згладжувальна функція вейвлет-функції  $\Psi(x)$  та  $S_{2^j}$  згладжувальний оператор, визначений як згортка:

$$S_{2^j} f(u) = \left( f(x) * \phi_{2^j}(x) \right)(u) = \int_{-\infty}^{\infty} f(x) \frac{1}{2^j} \phi\left(\frac{x-u}{2^j}\right) dx, \quad (2.20)$$

причому  $\phi_{2^j}(x)$  визначений як

$$\phi_{2^j}(x) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{x}{2^j}\right), \quad (2.21)$$

тоді, з вищезгаданого визначення, можна відмітити, що більше деталей  $f(x)$  видалені  $S_{2^j}$ , коли масштаб  $2^j$  стає більшим.

Якщо оригінальний сигнал є дискретною послідовністю  $f = (f(n))_{n \in Z}$  з кінцевою енергією, можна припустити, що фрагменти  $a_0(n)$  вхідного дискретного

сигналу не зовсім дорівнюють  $f(n)$ , але локальне середнє  $f$  по сусідству  $t = n$ . Таким чином,  $a_0(n)$  може бути записане формулою

$$a_0(n) = \langle f(t), \phi(t-n) \rangle = \int_{-\infty}^{\infty} f(t) \phi(t-n) dt. \quad (2.22)$$

Для кожного  $j > 0$  можна відмітити

$$S_{2^j} f = a_j(n) = \langle f(t), \phi_{2^j}(t-n) \rangle = \int_{-\infty}^{\infty} f(t) \phi_{2^j}(t-n) dt. \quad (2.23)$$

Діадне вейвлет-перетворення

$$W_{2^j} f = d_j(n) = Wf(n, 2^j) = \langle f(t), \psi_{2^j}(t-n) \rangle. \quad (2.24)$$

Для кожного масштабу  $2^j$  послідовність дискретних сигналів  $\left( S_{2^j} f, (W_{2^j} f)_{1 \leq j \leq J} \right)$  є дискретним діадним вейвлет-перетворенням  $f = (f(n))$ .

Отже, можна визначити діадне вейвлет-перетворення  $(S_2 f, W_2 f)$  як  $(Sf, Wf)$  для функції  $f(n)$ , де  $Sf$  виражає низькочастотну інформацію  $f(n)$ , а  $Wf$  – для високочастотної інформації  $f(n)$ .

## 2.5 Висновки до розділу

Вейвлет перетворення дає досить гарні результати для сприйняття людським оком, а також є досить легким в обчисленнях і не перевищує складність обчислень дискретного косинусного перетворення.

Перед модифікацією не обхідно дослідити вплив кількості коефіцієнтів усереднення на якість зображення. На основі результатів модифікувати алгоритм стиснення JPEG.

## РОЗДІЛ 3

### РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДИФІКОВАНОГО АЛГОРИТМУ JPEG

#### 3.1 Модифікований метод вейвлет перетворення

Класичне вейвлет-перетворення, що застосовується у алгоритмах стиснення зображень, працює тільки з одним коефіцієнтом усереднення. Це спричиняє необхідність використовувати додаткові попередні методи для зведення інформації про піксель зображення в системі RGB до одного значення. Також, для зворотного перетворення потрібно зберігати як саме зводилося це число, і за якими формулами його можна повернути [2].

Розглянемо результати вейвлет-перетворень (з 90% втрати інформації) починаючи з 2х коефіцієнтів усереднення (рисунки 3.1 – 3.6). Для їх оцінки використаємо критерії наступні якості:

- середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу

$$d(x, y) = \sqrt{\frac{\sum_{x,y}^{M,N} (f(x,y) - \hat{f}(x,y))^2}{M \cdot N}} \quad (3.1)$$

- критерій якості PSNR

$$d(x, y) = 10 \log_{10} \frac{255^2 \cdot M \cdot N}{\sum_{x,y}^{M,N} (f(x,y) - \hat{f}(x,y))^2} \quad (3.2)$$

Зображення з трьома коефіцієнтами мають більш плавні переходи та чіткіші деталі, ніж у зображення з двома коефіцієнтами. Те ж саме можна сказати про зображення з чотирма коефіцієнтами. Зображення, що мають більше чотирьох коефіцієнтів усереднення, для людського ока не відрізняються від

зображень з чотирма коефіцієнтами усереднення. Також, спираючись на значення коефіцієнтів якості (таблиця 3.1, рисунок 3.7 – 3.8) бачимо, що немає сенсу брати більше, ніж 4 коефіцієнти усереднення. Отже, для модифікації алгоритму JPEG обрано вейвлет-перетворення з чотирма коефіцієнтами усереднення.



Рисунок 3.1 – Вейвлет-перетворення з 2-ма коефіцієнтами усереднення



Рисунок 3.2 – Вейвлет-перетворення з 3-ма коефіцієнтами усереднення



Рисунок 3.3 – Вейвлет-перетворення з 4-ма коефіцієнтами усереднення



Рисунок 3.4 – Вейвлет-перетворення з 5-ма коефіцієнтами усереднення





Рисунок 3.5 – Вейвлет-перетворення з 6-ма коефіцієнтами усереднення



Рисунок 3.6 – Вейвлет-перетворення з 7-ма коефіцієнтами усереднення  
 Таким чином коефіцієнти перетворення будуть мати вигляд:

$$a_{i,j} = \frac{x_{8j+i} + x_{8j+(i+4)}}{\sqrt{2}}, \quad j = 1, \dots, 2^{n-1} - 1, i = 0 \dots 3 \quad (3.3)$$

$$d_{i,j} = \frac{x_{8j+i} - x_{8j+(i+4)}}{\sqrt{2}}, \quad j = 1, \dots, 2^{n-1} - 1, i = 0 \dots 3 \quad (3.4)$$

Можемо спростити метод до одномірного випадку, в результаті чого отримаємо матрицю наступного вигляду:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ d_{0,0} & d_{1,0} & d_{2,0} & d_{3,0} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ d_{0,n-1} & d_{1,n-1} & d_{2,n-1} & d_{3,n-1} \end{bmatrix} \quad (3.5)$$

В класичному вейвлет-перетворенні після отримання коефіцієнту усереднення та коефіцієнтів різниці, частина других відкидається. Це зменшує обсяг файлу, але збільшує втрати графічної інформації при зворотному перетворенні.

### 3.2 Модифікований алгоритм JPEG

Спочатку виконується перехід від кольорової схеми RGB до YCbCr наступним чином:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.578 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (3.6)$$

Далі виконується модифіковане вейвлет перетворення з коефіцієнтами наведеними вище (3.1) та (3.2).

Таблиця 3.1 – Значення критеріїв якості

|                        | Кількість коефіцієнтів усереднення |        |        |        |        |        |        |
|------------------------|------------------------------------|--------|--------|--------|--------|--------|--------|
|                        | 2                                  | 3      | 4      | 5      | 6      | 7      | 8      |
| Середньокв. відхилення | 1.0051                             | 1.0046 | 1.0046 | 1.0046 | 1.0046 | 1.0046 | 1.0046 |
| PSNR                   | 0.0779                             | 0.0739 | 0.0736 | 0.736  | 0.736  | 0.736  | 0.736  |

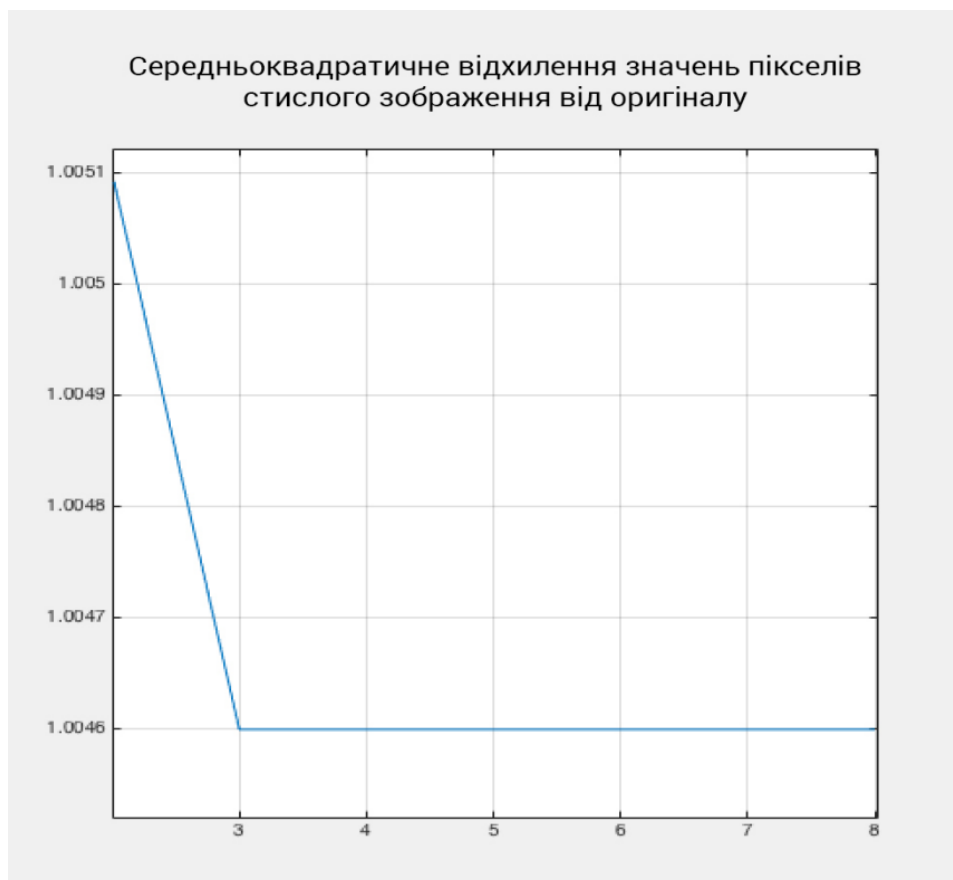


Рисунок 3.7 – Середньоквадратичне відхилення значень пікселів стиснутого зображення від оригіналу

Отримавши матрицю виду (3.3) виконуємо квантування коефіцієнтів. Коефіцієнти діляться на заздалегідь задане число – ступінь стиснення. При збільшенні цього числа знижується динамічний діапазон коефіцієнтів, вони стають ближче до 0, і ми отримуємо великий ступінь стиснення.

Після проведення квантування проводиться “зіг-заг” сканування та застосовується метод однопрохідного кодування Хаффмана.

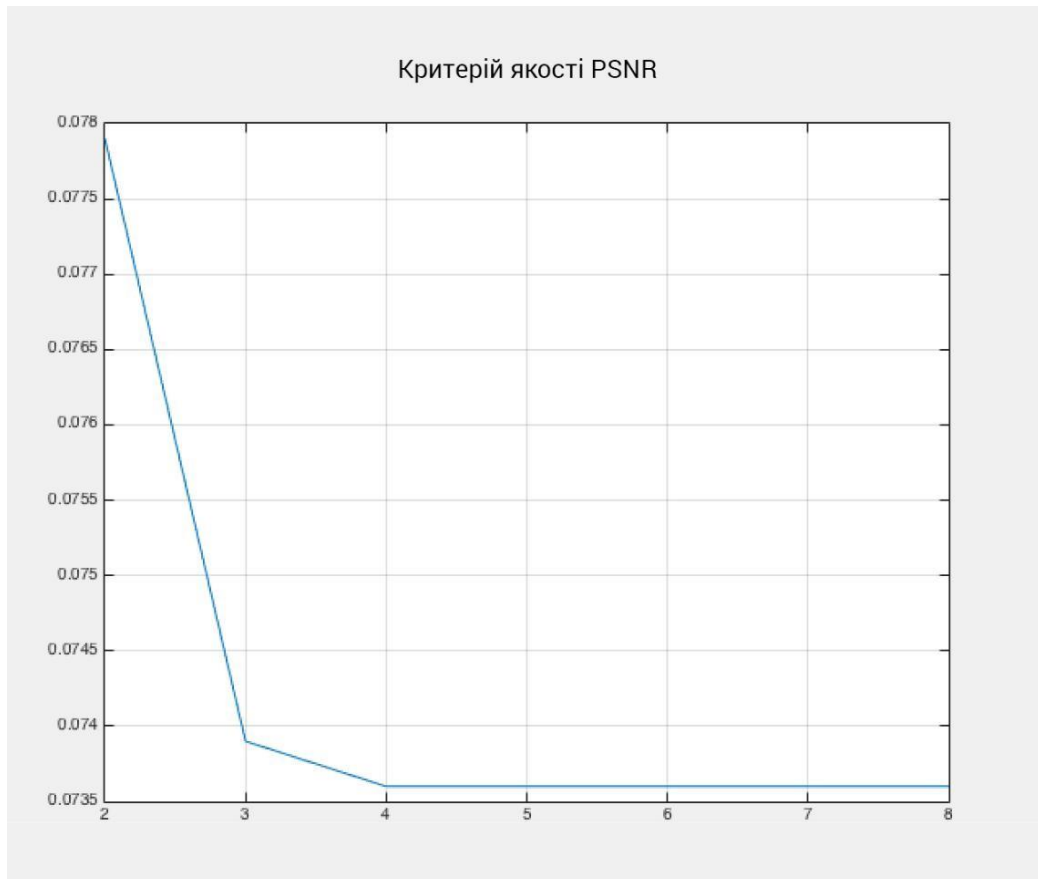


Рисунок 3.8 – Критерій якості PSNR

### 3.3 Обґрунтування вибору мови програмування

У якості мови програмування було обрано мову C++ та середовище програмування Microsoft Visual Studio 2019, що надало можливість створити програму зі зручним інтерфейсом користувача.

Перевагами C++ є:

- зручна об'єктно-орієнтована мова програмування, що дає змогу використовувати сучасні методи проектування (такі, як шаблони проектування) у відповідності до сучасних вимог до проектування програмного забезпечення;
- візуальні засоби розробки інтерфейсу;
- зручний доступ до багатих можливостей бібліотеки програмування тривимірної графіки.

Програма має досить досконалу об'єктну модель, що робить її більш гнучкою і дає можливість швидко додавати механізми побудови поверхонь за новими методами, якщо в цьому виникне потреба [27].

Microsoft Visual Studio – одне з найпоширеніших у нашій країні середовище розробки ПЗ. Сьогодні це основний і найбільш ефективний засіб розробки рішень для платформи Microsoft.

Visual Studio 2019 уособлює собою представлення корпорації Microsoft про інтелектуальні клієнтські додатки, які здатні забезпечити найширші можливості для роботи користувачів, і дозволяє швидко створювати та підключаються до баз даних програми. Visual Studio 2019 дає можливість дуже просто збирати та аналізувати, що сприяє прийняттю ефективних рішень. За допомогою Visual Studio 2019 можна швидко створювати більш безпечні, керовані та надійні додатки.

Visual Studio 2019 продовжує традиції, закладені попередніми версіями продукту, і містить багато новинок, призначених як для спрощення створення додатків для різних платформ окремими, так і для підвищення продуктивності всієї проектною команди.

Новинки, що з'явилися в Microsoft Visual Studio 2019, можна розділити на три категорії:

- підвищення продуктивності розробників;
- підтримка новітніх технологій;
- управління всім циклом створення додатків.

До новинок в області підвищення продуктивності розробників можна віднести розширення в мовах програмування, підтримку написання коду для декількох версій.

З допомогою засобів, включених до складу Visual Studio 2019, можна створювати додатки для широкого спектра платформ (розумних пристроїв (smart personal objects), стільникових телефонів, планшетних комп'ютерів, настільних комп'ютерів і серверів).

В Visual Studio 2019 є можливість орієнтуватися на різні версії. NET Framework в одному і тому ж середовищі розробки. Існує можливість створювати додатки для .NET Framework 2.0, 3.0, 3.5, 4.0, 4.5 тощо тобто підтримувати безліч проектів в одному середовищі.

Visual Studio 2019 містить нові засоби розробки, що полегшують створення додатків, що підключаються на найсучасніших платформах.

Visual Studio 2019 пропонує удосконалені засоби, які допомагають налагодити спільну діяльність в групах, включаючи інструменти для об'єднання в процесі розробки фахівців з баз даних і дизайнерів [35].

### **3.4 Методика порівняння**

Для проведення розрахунків було обрано загальнодоступну реалізацію алгоритму JPEG [9] мовою C++. Описані вище модифікації було реалізовано у коді, який надано у додатку В. Модифікація JPEG торкається лише етапу ентропійного кодування кодека baseline JPEG, що не впливає на якість стисненого зображення. Тому зображення, які отримують кодеками baseline JPEG і модифікованого JPEG, ідентичні за якістю, але відрізняються ступенем стиснення.

Інші режими роботи кодека JPEG (progressive, optimized, arithmetic, lossless) є надбудовами над baseline JPEG. Тому порівняння цих режимів роботи кодека з модифікованим JPEG даватиме такий самий результат, що і

для baseline JPEG, але з урахуванням виграшу в стиску, який був отриманий в ході експериментів. Порівняння проводилося на 24 різних чорно-білих зображення з бази KODAK [10]. Кожне зображення бази представлене у чотирьох розмірах: 0.4, 0.6, 1.6 та 6.3 мегапікселя. Для оцінки якості стислих зображень використовувалися спеціальні метрики адекватні людському сприйняттю. Відомо близько 20 альтернативних математичних критеріїв оцінки якості спотвореного зображення, зокрема широко поширений критерій PSNR, який не враховує особливості людського зору. З результатів попередніх робіт [11] слід, що з оцінки якості зображення найкращим є математичний критерій VIF [12]. У ході експерименту проводилося кодування тестових зображень кодеками baseline JPEG та модифікований JPEG. Параметр якості  $p$  змінювався у всьому діапазоні допустимих значень. Для кожного тестового зображення обчислювалися ступінь стиснення як відношення розміру вихідного файлу до стиснутого, і якість за критерієм VIF [32] (рисунки 3.9-3.10).

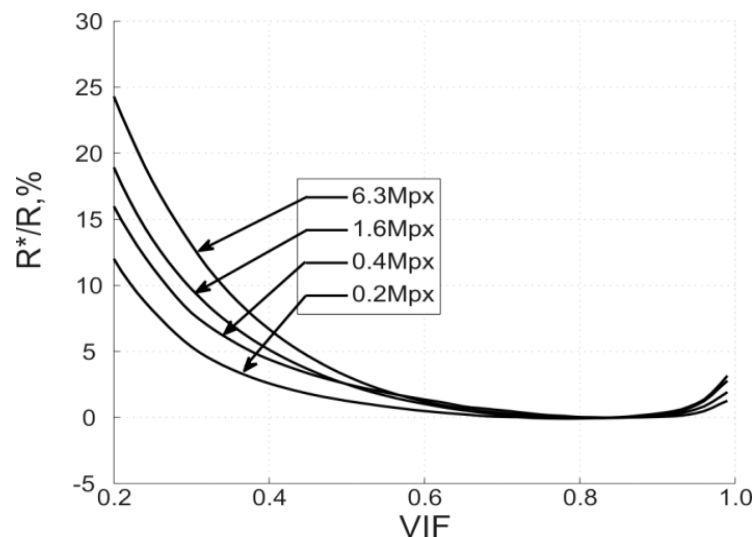


Рисунок 3.9 – Середній виграш у стисненні модифікованим JPEG за 24 зображеннями; значення біля кривих – розмір зображення у мегапікселях



Рисунок 3.10 – Зображення №21 з бази KODAK після компресії для різних рівнів якості VIF (рядково зліва направо: 0.11, 0.21, 0.31, 0.46, 0.68 та 0.99)

Результат стиснення зображення з рисунку 3.10 представлений у додатку Б таблиці Б.1. Для кожного рівня якості наведено бітові витрати на піксель при використанні кодека JPEG та модифікованого JPEG. В останньому рядку таблиці наведено процентний приріст у стисканні запропонованої модифікації проти стандартного кодека. Виграш змінюється від  $\sim 0,4\%$  до  $20,4\%$  в залежності від рівня якості стисненого зображення.

Аналіз рисунку 3.9 та таблиці Б.1 дозволяє зробити такі висновки:

- У разі застосування запропонованих модифікацій виграш у стиску є у всьому діапазоні якості. За хорошої якості він складає  $\sim 0,5\%$ , при високому до  $\sim 4,5\%$ , а при середньому і поганому зростає до  $\sim 20\%$ , в залежності від розміру зображення.

- При роботі із зображеннями розміром від 0,2 до 6,3 Мп виграш у стиску змінюється відповідно: від 2,0 до 4,0% у діапазоні середньої якості, від 5,0 до 20,0% у діапазоні поганої якості.



- З використанням інших таблиць Хаффмана, оптимізованих іншого рівня якості, форма кривих на рисунку 3 збережеться. При цьому мінімум зміститься в область якості, для якої оптимізація [33].

### 3.5 Аналіз результатів роботи модифікованого алгоритму

Порівняємо різні ступені стиснення зображень за допомогою алгоритму JPEG та модифікованого JPEG. На рисунках 3.11 – 3.12 наведено зображення з 50% коефіцієнтом стиснення.

Як бачимо, зображення стиснуте за допомогою модифікованого алгоритму мало відрізняється від вихідного зображення, в той час як на зображенні стиснутому за допомогою оригінального алгоритму добре видно спотворення. Критерії якості наведені у таблиці 3.1 та їх ілюстрація на рисунках 3.13 – 3.17.



Рисунок 3.11 – Вихідне зображення

Розглянемо результати більшого ступеню стиснення (рисунок 3.13 – 3.17).

При ступеню стиснення у 80% оригінальний алгоритм дає у результаті значні спотворення зображення. Модифікований алгоритм також дає спотворення, але набагато менші.



Рисунок 3.12 – Зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 50%



Рисунок 3.13 – Зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 50%



Рисунок 3.14 – Зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 80%



Рисунок 3.15 – Зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 80%

При стисненні в 90% (рисунки 3.16 – 3.17) модифікований алгоритм також дає значні спотворення зображення, але значно прийнятні для ока, ніж у результаті стиснення оригінальним алгоритмом. Також модифікований алгоритм дає можливість позбутися розбиття зображення на квадрати.

Розглянемо як алгоритми впливають на розміри файлів таблиця 3.2. Як бачимо, модифікований алгоритм дає змогу отримати менший розмір файлу, ніж оригінальний алгоритм [34].



Рисунок 3.16 – Зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 90%



Рисунок 3.17 – Зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 90%

Початковий розмір файлу складає 84 КВ.

У додатку Б таблиці Б.3 наведені значення критеріїв якості.

На рисунку 3.18 наведена залежність розміру файлу зображення від коефіцієнта стиснення.

У додатку Б на рисунку Б.4 наведено середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу. У додатку Б на рисунку Б.5 наведено порівняння за критерієм якості PSNR.

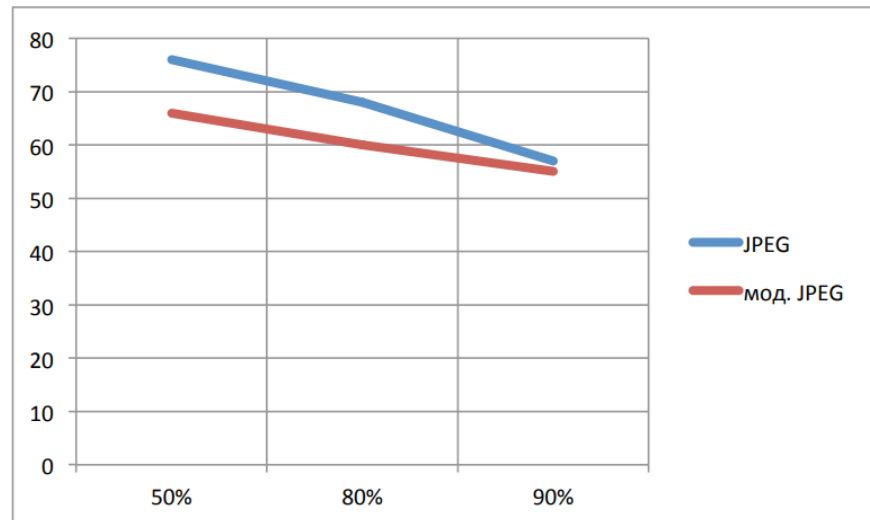


Рисунок 3.18 – Залежність розміру файлу зображення від коефіцієнта стиснення

### 3.6 Висновок до розділу

Проведено експериментальне дослідження впливу кількості коефіцієнтів усереднення на якість зображення при вейвлет-перетворенні. За результатами дослідження обрано 4 коефіцієнти. На основі цього запропонована модифікація алгоритму стиснення зображень JPEG.

Проведено аналіз результатів оригінального та модифікованого алгоритмів стиснення зображень JPEG. Модифікований алгоритм показує значну перевагу. Оптимальним відношенням якість/розмір файлу є стиснення зображення до 80%.

## 4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

### 4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями [31].

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [32].

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

| Критерії  | Експерт (ПІБ, посада) |    |    |
|---|-----------------------|----|----|
|   | 1                     | 2  | 3  |
|   | Бали:                 |    |    |
| 1. Технічна здійсненність концепції                         | 5                     | 4  | 4  |
| 2. Ринкові переваги (наявність аналогів)                    | 1                     | 2  | 1  |
| 3. Ринкові переваги (ціна продукту)                         | 4                     | 4  | 3  |
| 4. Ринкові переваги (технічні властивості)                  | 3                     | 3  | 2  |
| 5. Ринкові переваги (експлуатаційні витрати)                | 2                     | 2  | 2  |
| 6. Ринкові перспективи (розмір ринку)                       | 2                     | 2  | 2  |
| 7. Ринкові перспективи (конкуренція)                        | 4                     | 4  | 3  |
| 8. Практична здійсненність (наявність фахівців)             | 5                     | 5  | 5  |
| 9. Практична здійсненність (наявність фінансів)             | 3                     | 4  | 3  |
| 10. Практична здійсненність (необхідність нових матеріалів) | 4                     | 5  | 5  |
| 11. Практична здійсненність (термін реалізації)             | 4                     | 4  | 4  |
| 12. Практична здійсненність (розробка документів)           | 4                     | 3  | 4  |
| Сума балів  | 41                    | 42 | 38 |
| Середньоарифметична сума балів $CB_c$                       | 40,3                  |    |    |

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» становить 40,3 бала, що, відповідно [32], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

## 4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розрахуємо за формулою [33]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом

і при цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$ ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога

| Показники (параметри)   | Одиниця вимірювання | Аналог | Проектований продукт | Відношення параметрів нової розробки до аналога | Питома вага показника |
|---|---------------------|--------|----------------------|---|-----------------------|
| Коефіцієнт стиснення  | %                   | 60     | 80                   | 1,33  | 0,35                  |
| Якість зображення за критерієм PSNR (в відсотках від оригіналу) | %                   | 83     | 92                   | 1,11  | 0,25                  |
| Об'єм оперативної пам'яті, який займає програма                 | МБ                  | 256    | 128                  | 2   | 0,15                  |
| Об'єм програмного забезпечення на диску                         | МБ                  | 150    | 50                   | 3   | 0,1                   |
| Швидкість обробки зображення (1920x1080 pсі)                    | с                   | 1      | 0,5                  | 2   | 0,15                  |



Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,33 \cdot 0,35 + 1,11 \cdot 0,25 + 2 \cdot 0,15 + 3 \cdot 0,1 + 2 \cdot 0,15 = 1,64.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,64 рази.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

#### 4.3.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [32]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=24$  дні.

$$Z_o = 17400,00 \cdot 24 / 24 = 17400,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.3 – 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  | 17400,00                      | 725,00                      | 24                | 17400,00                        |
| Науковий співробітник   | 17150,00                      | 714,58                      | 14                | 10004,17                        |
| Інженер-програміст 1-ї категорії                                    | 17050,00                      | 710,42                      | 24                | 17050,00                        |
| Консультант-аналітик цифрових обчислюваних систем обробки зображень | 17200,00                      | 716,67                      | 15                | 10750,00                        |
| Технік  | 7000,00                       | 291,67                      | 6                 | 1750,00                         |
| Всього  |                               |                             |                   | 56954,17                        |

#### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

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

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

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

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де  $M_M$  – розмір мінімальної місячної заробітної плати, прийmemo  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення (таблиця Б.2, додаток Б) [32];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 24$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_i = 6700,00 \cdot 1,10 \cdot 1,65 / (24 \cdot 8) = 63,34 \text{ грн.}$$

$$З_{pl} = 63,34 \cdot 8,50 = 538,36 \text{ грн.}$$

Таблиця 4.4 – 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 |
|---|------------------------|---------------|---------------------|-------------------------------|----------------------------------|
| Встановлення допоміжного обладнання   | 8,50                   | 2             | 1,10                | 63,34                         | 538,36                           |
| Інсталяція програмного забезпечення   | 6,25                   | 3             | 1,35                | 77,73                         | 485,82                           |
| Встановлення цифрових обчислювальних систем обробки зображень                           | 5,20                   | 5             | 1,70                | 97,88                         | 508,99                           |
| Відлагодження інтерполяційних модулів графічної системи                                 | 5,50                   | 4             | 1,50                | 86,37                         | 475,02                           |
| Підготовка цифрової експериментальної моделі  | 8,00                   | 4             | 1,50                | 86,37                         | 690,94                           |
| Формування бази даних підсистеми формування зображень                                   | 14,00                  | 2             | 1,10                | 63,34                         | 886,70                           |
| Формування бази даних для підсистеми обробки зображень                                  | 12,00                  | 3             | 1,35                | 77,73                         | 932,77                           |
| Узгодження параметрів системи комп'ютерної графіки для формування оптимальних зображень | 2,50                   | 3             | 1,35                | 77,73                         | 194,33                           |
| Тестування програмного забезпечення   | 5,00                   | 3             | 1,35                | 77,73                         | 388,65                           |
| Всього  |                        |               |                     |                               | 5101,57                          |

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{доо}} = (Z_o + Z_p) \cdot \frac{H_{\text{доо}}}{100\%}, \quad (4.5)$$

де  $H_{\text{доо}}$  – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{доо}} = (56954,17 + 5101,57) \cdot 11 / 100\% = 6826,13 \text{ грн.}$$

#### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{доо}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.6)$$

де  $H_{\text{зн}}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (56954,17 + 5101,57 + 6826,13) \cdot 22 / 100\% = 15154,01 \text{ грн.}$$

#### 4.3.3 Сировина та матеріали

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.7)$$

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

$n$  – кількість видів матеріалів;

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

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 2,0 \cdot 225,00 \cdot 1,12 - 0 \cdot 0 = 504,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на матеріали

| Найменування матеріалу, марка, тип, сорт                      | Ціна за 1 кг, грн | Норма витрат, кг | Величина відходів, кг | Ціна відходів, грн/кг | Вартість витраченого матеріалу, грн |
|---|-------------------|------------------|-----------------------|-----------------------|-------------------------------------|
| Папір канцелярський офісний (А4)                              | 225,00            | 2,0              | 0                     | 0                     | 504,00                              |
| Папір для заміток (А5)  | 116,00            | 4,0              | 0                     | 0                     | 519,68                              |
| Начиння канцелярське  | 195,00            | 3,0              | 0                     | 0                     | 655,20                              |
| Органайзер офісний  | 183,00            | 3,0              | 0                     | 0                     | 614,88                              |
| Картридж для принтера   | 950,00            | 1,0              | 0                     | 0                     | 1064,00                             |
| Диск оптичний CD-RW   | 21,00             | 3,0              | 0                     | 0                     | 70,56                               |
| USB флеш накопичувач Transcend 16Gb JetFlash 700 (TS64GJF700) | 120,00            | 1,0              | 0                     | 0                     | 134,40                              |
| Всього  |                   |                  |                       |                       | 3562,72                             |

#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_6$ ), які використовують при проведенні НДР на тему «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» відсутні.

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.8)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань устаткування.

$$B_{\text{спец}} = 26750,00 \cdot 1 \cdot 1,11 = 29692,50 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.6 – Витрати на придбання спецустаткування по кожному виду

| Найменування устаткування                       | Кількість,<br>шт | Ціна за<br>одиницю, грн | Вартість,<br>грн |
|---|------------------|-------------------------|------------------|
| Мультимедійна проекційна система EPSON-230AJ-FX | 1                | 26750,00                | 29692,50         |
| Всього  |                  |                         | 29692,50         |

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{инрг}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (4.9)$$

де  $C_{\text{инрг}}$  – ціна придбання одиниці програмного засобу даного виду, грн;

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

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 3150,00 \cdot 1 \cdot 1,11 = 3496,50 \text{ грн.}$$

Отримані результати зведемо до таблиці:



Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

| Найменування програмного засобу  | Кількість, шт | Ціна за одиницю, грн | Вартість, грн |
|--|---------------|----------------------|---------------|
| Комп'ютерна програма для адаптивної нормалізації векторів у рядку растеризації | 1             | 3150,00              | 3496,50       |
| Прикладний пакет розробника Java   | 1             | 4100,00              | 4551,00       |
| Всього   |               |                      | 8047,50       |

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

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

$$A_{обл} = \frac{Ц_{б}}{T_{г}} \cdot \frac{t_{вик}}{12}, \quad (4.10)$$

де  $Ц_{б}$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

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

$T_{г}$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (21570,00 \cdot 1) / (2 \cdot 12) = 898,75 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

| Найменування обладнання                        | Балансова вартість, грн | Строк корисного використання, років | Термін використання обладнання, місяців | Амортизаційні відрахування, грн |
|--|-------------------------|-------------------------------------|---|---------------------------------|
| Програмно-аналітичний комплекс                 | 21570,00                | 2                                   | 1                                       | 898,75                          |
| Графічно-обчислювальний комплекс обробки даних | 25250,00                | 2                                   | 1                                       | 1052,08                         |
| Програмні продукти обробки графічних даних     | 15600,00                | 3                                   | 1                                       | 433,33                          |
| Обладнання виводу графічної інформації         | 9200,00                 | 4                                   | 1                                       | 191,67                          |
| Місце оператора спеціалізоване                 | 8620,00                 | 5                                   | 1                                       | 143,67                          |
| Офісна оргтехніка                              | 11400,00                | 4                                   | 1                                       | 237,50                          |
| Дослідницька лабораторія                       | 225000,00               | 25                                  | 1                                       | 750,00                          |
| Прикладний пакет Microsoft Office 2016         | 7900,00                 | 2                                   | 1                                       | 329,17                          |
| ОС Windows 10 pro                              | 8450,00                 | 2                                   | 1                                       | 352,08                          |
| Всього   |                         |                                     |   | 4388,25                         |

#### 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{\epsilon ni}}{\eta_i}, \quad (4.11)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; прийmemo  $C_e = 6,20$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

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

$$B_e = 0,32 \cdot 180,0 \cdot 6,20 \cdot 0,95 / 0,97 = 357,12 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.9 – Витрати на електроенергію

| Найменування обладнання                        | Встановлена потужність, кВт | Тривалість роботи, год | Сума, грн |
|--|-----------------------------|------------------------|-----------|
| Програмно-аналітичний комплекс                 | 0,32                        | 180,0                  | 357,12    |
| Графічно-обчислювальний комплекс обробки даних | 0,56                        | 180,0                  | 624,96    |
| Мультимедійна проекційна система               | 0,11                        | 180,0                  | 122,76    |
| Обладнання виводу графічної інформації         | 0,18                        | 11,0                   | 12,28     |
| Місце оператора спеціалізоване                 | 0,15                        | 180,0                  | 167,40    |
| Офісна оргтехніка                              | 0,75                        | 2,1                    | 9,77      |
| Всього   |                             |                        | 1294,28   |

### 4.3.9 Службові відрядження

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

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.12)$$

де  $H_{cv}$  – норма нарахування за статтею «Службові відрядження», приймемо  $H_{cv} = 20\%$ .

$$B_{cv} = (56954,17 + 5101,57) \cdot 20 / 100\% = 12411,15 \text{ грн.}$$

### 4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.13)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо  $H_{cn} = 30\%$ .

$$B_{cn} = (56954,17 + 5101,57) \cdot 30 / 100\% = 18616,72 \text{ грн.}$$

#### 4.3.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (4.14)$$

де  $H_{\text{ів}}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{\text{ів}} = 60\%$ .

$$I_{\text{в}} = (56954,17 + 5101,57) \cdot 60 / 100\% = 37233,44 \text{ грн.}$$

#### 4.3.12 Накладні (загальновиробничі) витрати

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.15)$$

де  $H_{\text{нзв}}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{\text{нзв}} = 100\%$ .

$$B_{\text{нзв}} = (56954,17 + 5101,57) \cdot 100 / 100\% = 62055,73 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + M + K_e + B_{\text{стел}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_e + B_{\text{нзв}}. \quad (4.16)$$

$$\begin{aligned} B_{\text{заг}} = & 56954,17 + 5101,57 + 6826,13 + 15154,01 + 3562,72 + 0,00 + 29692,50 + \\ & + 8047,50 + 4388,25 + 1294,28 + 12411,15 + 18616,72 + 37233,44 + 62055,73 = \\ & = 261338,16 \text{ грн.} \end{aligned}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.17)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,95$ .

$$ZB = 261338,16 / 0,95 = 275092,80 \text{ грн.}$$

#### 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Результати дослідження проведені за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$\Delta N$  – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

| Показник                              | 1-й рік | 2-й рік | 3-й рік | 4-й рік |
|---------------------------------------|---------|---------|---------|---------|
| Збільшення кількості споживачів, осіб | 3000    | 5500    | 8000    | 6000    |

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 25000 осіб;

$C_o$  – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 850,00 грн;

$\pm \Delta C_o$  – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 96,55 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [32]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.18)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 35\%$ ;

$\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році  $\vartheta = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (96,55 \cdot 25000,00 + 946,55 \cdot 3000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 1251412,41 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (96,55 \cdot 25000,00 + 946,55 \cdot 8500) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 2491539,63 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (96,55 \cdot 25000,00 + 946,55 \cdot 16500) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 4295361,03 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (96,55 \cdot 25000,00 + 946,55 \cdot 22500) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 5648227,09 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків  $ПП$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (4.20)$$



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

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

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau=0,26$ ;

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

$$\begin{aligned} \text{ПП} &= 1251412,41/(1+0,26)^1 + 2491539,63/(1+0,26)^2 + 4295361,03/(1+0,26)^3 + \\ &+ 5648227,09/(1+0,26)^4 = 993184,46 + 1569374,92 + 2147276,83 + 2240938,66 = \\ &= 6950774,87 \text{ грн.} \end{aligned}$$

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

$$PV = k_{инв} \cdot 3B, \quad (4.21)$$

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

$3B$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 275092,80 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 275092,80 = 550185,60 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.22)$$

де  $ПП$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 6950774,87 грн;

$PV$  – теперішня вартість початкових інвестицій, 550185,60 грн.

$$E_{абс} = ПП - PV = 6950774,87 - 550185,60 = 6400589,26 \text{ грн.}$$

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

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.23)$$

де  $E_{абс}$  – абсолютний економічний ефект вкладених інвестицій, 6400589,26 грн;

$PV$  – теперішня вартість початкових інвестицій, 550185,60 грн;

$T_{ж}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 6400589,26/550185,60)^{1/4} - 1 = 0,89.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{мін}$

$$\tau_{\min} = d + f, \quad (4.24)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = 0,12$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,37.

$\tau_{\min} = 0,12 + 0,37 = 0,49 < 0,89$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.25)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,89 = 1,13 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### 4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення» становить 40,3 бала, що, свідчить про комерційну

важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,64 рази.

Також термін окупності становить 1,13 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення».

## ВИСНОВКИ

У даній магістерській роботі було розглянуті методи стиснення зображень, розглянуті методи стиснення без втрат та з втратами, проведено їх аналіз та порівняння. На основі аналізу виявлено, що можливо покращити алгоритм стиснення JPEG.

Запропоновано модифікацію алгоритму JPEG, що дає вигоду у стиску до 20% порівняно з базовим кодеком JPEG залежно від необхідної якості та вихідного розміру зображення. Модифікація дає вигоду у стиску як на зображеннях з дрібними деталями, так і на фонових зображеннях.

Модифікація не збільшує трудомісткість базового алгоритму JPEG при компресії та декомпресії зображень. Тому цю зміну можна впроваджувати в існуючі програмні та апаратні модулі стиснення зображення на основі JPEG.

Розроблено та програмно реалізований алгоритм стиснення цифрових зображень, який дозволив підвищити ступінь стискання зображень та зберегти прийнятну якість. Запропонований метод може використовуватись для пришвидшення завантаження сайтів у мережі Інтернет. Можливе перенесення модифікації алгоритму JPEG у відеокодеки на основі JPEG, наприклад Motion-JPEG.

В економічній частині було розраховано:

- рівень комерційного потенціалу розробки становить 40,3 бала;
- узагальнений коефіцієнт якості розробки у порівнянні з існуючими аналогами приблизно в 1,64 рази більший;
- термін окупності становить 1,13 р.

Проведені розрахунки свідчать про економічну ефективність та доцільність провадження нового програмного продукту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волошина А.О. Модифікація алгоритму стиснення JPEG / А. О. Волошина, В. В. Гармаш: LI Науково-технічна конференція факультету комп'ютерних систем і автоматики (2022). – URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15276/12923>
2. Основи цифрової графіки та кольору: веб-сайт. URL: <http://www.hitechforum.ru/index.php>
3. Уелстід С.: Фрактали та вейвлети для стиснення зображень у дії: посібник. Москва, 2003. 319 с.
4. Крилов Е.В., Анікін В.К., Анікіна Е.В.: Дослідження вейвлетного методу стиснення зображень для підвищення швидкодії веб-додатків: посібник. Москва, 2013. с. 35-40
5. JPEG Compression. Вікіпедія - веб-сайт. URL: [http://www.fileformat.info/mirror/egff/ch09\\_06.htm](http://www.fileformat.info/mirror/egff/ch09_06.htm)
6. Damerval C. A fast algorithm for bidimensional EMD / C. Damerval, S. Meignen, V. Perrier : IEEE Signal Processing Letters. 2015. V. 12(10). с.701-704.
7. Гармаш В.В. Метод стиснення цифрових зображень на основі усічення простору вейвлет-коефіцієнтів / В. В. Гармаш, Н. М. Ольшанська: XLVIII Науково-технічна конференція факультету комп'ютерних систем і автоматики (2019). URL : <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2019/paper/view/6617/5488>
8. Гармаш В.В. Метод фільтрації цифрових кольорових зображень / В. В. Гармаш, С. В. Пучко [Електронний ресурс]: XLIX Науково-технічна конференція факультету комп'ютерних систем і автоматики (2020). – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/8810/7864>

9. Барченко К. В. Аналіз методів фільтрації зображень / Барченко К. В., Білошкурський С. С., Гармаш В. В. // Вісник Хмельницького національного університету. – 2012. – № 4. – С. 79. – ISSN 2226-9150.
10. Shevchuk V.V. Color image denoising using “chanell method” / Shevchuk V.V. Garmash V. V. // Vdecky pokrok na prelomu tysyachalety – 2011. – Р. 3 – 6. – ISBN 978-966-8736-05-6.
11. Гармаш В.В. Метод фільтрації кольорових зображень на основі моделі YCrCb/ В.В. Гармаш // Вісник Черкаського технологічного університету. Серія "Технічні науки". – 2012. – №3. – С. 69–73. – ISSN 2306-4412.
12. Барабан М. В. Метод фільтрації цифрових кольорових зображень, отриманих в умовах недостатнього рівня освітлення [Текст] / М. В. Барабан, О. М. Бевз, В. В. Гармаш // Оптико-електронні інформаційно-енергетичні технології. – 2017. – № 2. – С. 5-10.
13. Coifman R.R., Meyer Y., Wickerhauser V. Wavelet Analysis and Signal Processing Wavelets // Wavelets and their Applications. Boston: Jones and Barlett. 2014. с. 153-178.
14. Daubechies L, Sweldens W. Factoring Wavelet Transforms into Lifting Steps : IEEE Trans, on Image Processing. 2010. Vol. 9. No. 3. с. 480-496.
15. Goutsias J. Mathematical Morphology and Its Application to Image and Signal Processing : J. Goutsias, L. Vincent, D.S. Bloomberg. Boston: Kluwer Academic Publishers. 2019. 255 с.
16. Wallace G.K. JPEG algoritm for image compression standard : Communications of the ACM. 1991. Vol. 34. №4. с. 30-44.
17. Умняшкін С.В., Космач М.В. Оптимізація кодування цифрових зображень методом JPEG : Вісті вузів. Електроніка. 2000. №4-5. с. 139-141.
18. Умняшкін С. В. Використання контекстного арифметичного кодування для підвищення стиснення даних за схемою JPEG : Вісті вузів. Електроніка. 2001. № 3. с. 96-99.

19. Умняшкін С.В., Куріна В.В. Алгоритм стиснення зображень на основі дискретного псевдокосинусного перетворення : Цифрова обробка сигналів. 2009. №3. с. 2-7.
20. Jie Liang, Trac D. Tran. Fast multiplierless approximations of the DCT with lifting scheme : IEEE transactions on signal processing. 2017. Vol. 49. №12. с. 3032 – 3044.
21. Голованов Р.В., Каліткін Н.М. Продовження підвищеної гладкості для фур'є-апроксимації неперіодичної функції : Доповіді Академії наук. 2012. с. 25-29.
22. Fenwick P. Punctured Elias Codes для variable-length coding of the integers : Technical Report 137. Department of Computer Science. The University of Auckland. 1996. с. 123
23. Беляєв І.А. СФ-блок кодування Хаффмана для стиснення зображень за стандартом JPEG : Проблеми розробки перспективних мікро- та наноелектронних систем. 2010. с. 332-335.
24. JPEG C++ source code. URL: <http://www.infai.org/jpeg/>. - 08.12.22
25. Kodak Lossless True Color Image Suite.  
URL: <http://r0k.us/graphics/kodak/>. - 07.12.2022.
26. Голованов Р.В., Каліткін Н.М. Статистична недостовірність поширених критеріїв оцінки якості спотвореного зображення : Цифрова обробка сигналів. 2013. №3. с. 74-79.
27. Sheikh H. R., Bovik A. C. Image Information and Visual Quality : IEEE Transactions on Image Processing. 2006. Vol. 15. с. 430-444.
29. Johnson B. Professional Visual Studio 2017 / B. Johnson. – Wrox, 2018. с. 864
30. Методичні вказівки до виконання магістрських дипломних робіт (проектів) для студентів спеціальностей 126 – «Інформаційні системи та технології», 151 – «Автоматизація та комп'ютерно-інтегровані технології» : Уклад. Р. Н. Кветний, О. М. Бевз, О. В. Бісікало. Вінниця : ВНТУ, 2019. 26 с.



31. Методичні рекомендації з комерціалізації розробок, створених в результаті науково-технічної діяльності – К.: Наказ Державного комітету України з питань науки, інновацій та інформатики (Лист № 1/06-4-97 від 13.09.2010 р.).

32. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

33. 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  
(обов'язковий)  
Технічне завдання

ЗАТВЕРДЖЕНО  
Зав. кафедри АІТ  
\_\_\_\_\_ Бісікало О.В.  
«\_\_» \_\_\_\_\_ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ  
на магістерську кваліфікаційну роботу  
«Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-  
перетворення»  
08-02.МКР.004.00.000 ТЗ

Керівник роботи:  
д.т.н., проф. Бісікало О.В.  
«\_\_» \_\_\_\_\_ 2022 р.

Виконавець: ст. гр. ІСТ-21м  
Конопльов Е.Ю.  
«\_\_» \_\_\_\_\_ 2022 р.

### 1. Назва та галузь застосування

Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення. Розроблене програмне забезпечення застосовуватиметься в системах обробки та передавання зображень.

### 2. Підстава для розробки

Розробку системи здійснювати на підставі наказу по університету № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 2022 р. та завдання до магістерської кваліфікаційної роботи, складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій».

### 3. Мета та призначення розробки

Метою даної роботи є підвищення коефіцієнту стиснення цифрових зображень шляхом модифікації алгоритму стиснення JPEG, що дозволяє ефективно стискати зображення при мінімальному спотворенні.

Призначенням розробки є створення програмного забезпечення, яке би реалізовувало модифікований алгоритм стиснення JPEG на основі вейвлет-перетворення.

### 4. Джерела розробки

1. Волошина А.О. Модифікація алгоритму стиснення JPEG / А. О. Волошина, В. В. Гармаш: LI Науково-технічна конференція факультету комп'ютерних систем і автоматики (2022). – URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15276/12923>

2. Основи цифрової графіки та кольору: веб-сайт. URL: <http://www.hitechforum.ru/index.php>

3. Уелстід С.: Фрактали та вейвлети для стиснення зображень у дії: посібник. Москва, 2003. 319 с.

4. Damerval C. A fast algorithm for bidimensional EMD / C. Damerval, S. Meignen, V. Perrier : IEEE Signal Processing Letters. 2015. V. 12(10). с.701-704.

5. Гармаш В.В. Метод стиснення цифрових зображень на основі усічення простору вейвлет-коефіцієнтів / В. В. Гармаш, Н. М. Ольшанська: XLVIII Науково-технічна конференція факультету комп'ютерних систем і автоматики (2019). URL : <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2019/paper/view/6617/54885>.

6. Coifman R.R., Meyer Y., Wickerhauser V. Wavelet Analysis and Signal Processing Wavelets // Wavelets and their Applications. — Boston: Jones and Barlett, 2014. — Pp. 153-178.

#### 5. Показники призначення

Вихідні дані для роботи програми:

- Кольорові та півтонові цифрові зображення;
- мінімальна роздільна здатність зображення 128 x 128 пікселів
- максимальна роздільна здатність зображення 2048 x 2048 пікселів;
- коефіцієнт стиснення – 3 – 20%.

Результати роботи програми:

- PSNR обробленого зображення > 20 Дб.

#### 6. Економічні показники

До економічних показників входять:

- рівень комерційного потенціалу розробки становить 40,3 бала;
- узагальнений коефіцієнт якості розробки у порівнянні з існуючими аналогами приблизно в 1,64 рази більший;
- термін окупності становить 1,13 р.

## 7. Стадії розробки

1. Розділ 1 «Аналіз сучасних методів стиснення зображень» має бути виконаний до «20» жовтня 2022 р.

2. Розділ 2 «Застосування ортогональних перетворень для стиснення зображень» має бути виконаний до «01» листопада 2022 р.

3. Розділ 3 «Розробка та реалізація модифікованого алгоритму JPEG» має бути виконаний до «15» листопада 2022 р.

4. Економічний розділ має бути виконаний до «20» листопада 2022 р.

## 8. Порядок контролю та приймання

1. Рубіжний контроль. Провести до «01» листопада 2022 р.

2. Попередній захист магістерської кваліфікаційної роботи. Провести до «07» грудня 2022 р.

3. Захист магістерської кваліфікаційної роботи. Провести в період з \_\_\_\_\_ до \_\_\_\_\_ р.

Додаток Б  
(обов'язковий)  
Графічна частина

Зав. кафедри АІТ \_\_\_\_\_ д-р техн. наук, професор каф. АІТ  
Бісікало О.В.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Науковий керівник \_\_\_\_\_ д-р техн. наук, професор каф. АІТ  
Бісікало О.В.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Тех. контроль \_\_\_\_\_ д-р техн. наук, професор каф. АІТ  
Бісікало О.В.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Нормоконтроль \_\_\_\_\_ д-р техн. наук, професор каф. АІТ  
Бісікало О.В.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Рецензент \_\_\_\_\_ д-р техн. наук, професор каф. КСУ  
Ковтун В.В.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Студент гр. ІСТ-21м \_\_\_\_\_ Конопльов Е.Ю.  
(підпис) (ініціали та прізвище)

## Продовження додатка Б

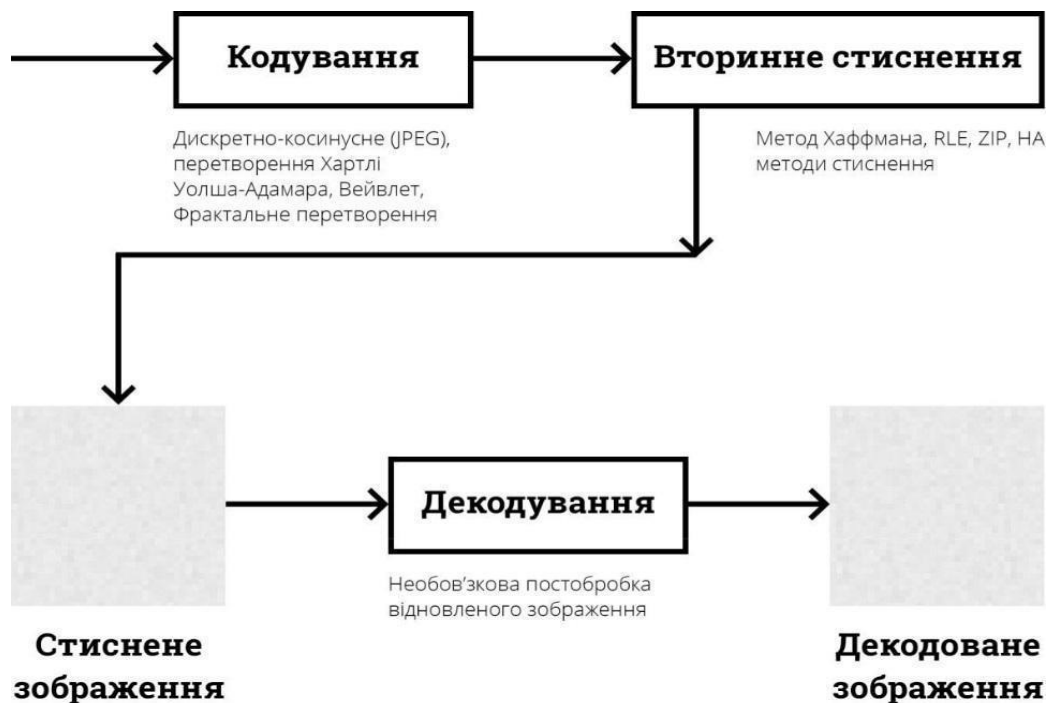


Рисунок Б.1 – Основні етапи стиснення цифрових зображень





Рисунок Б.2 – Схема стиснення зображення по стандарту JPEG

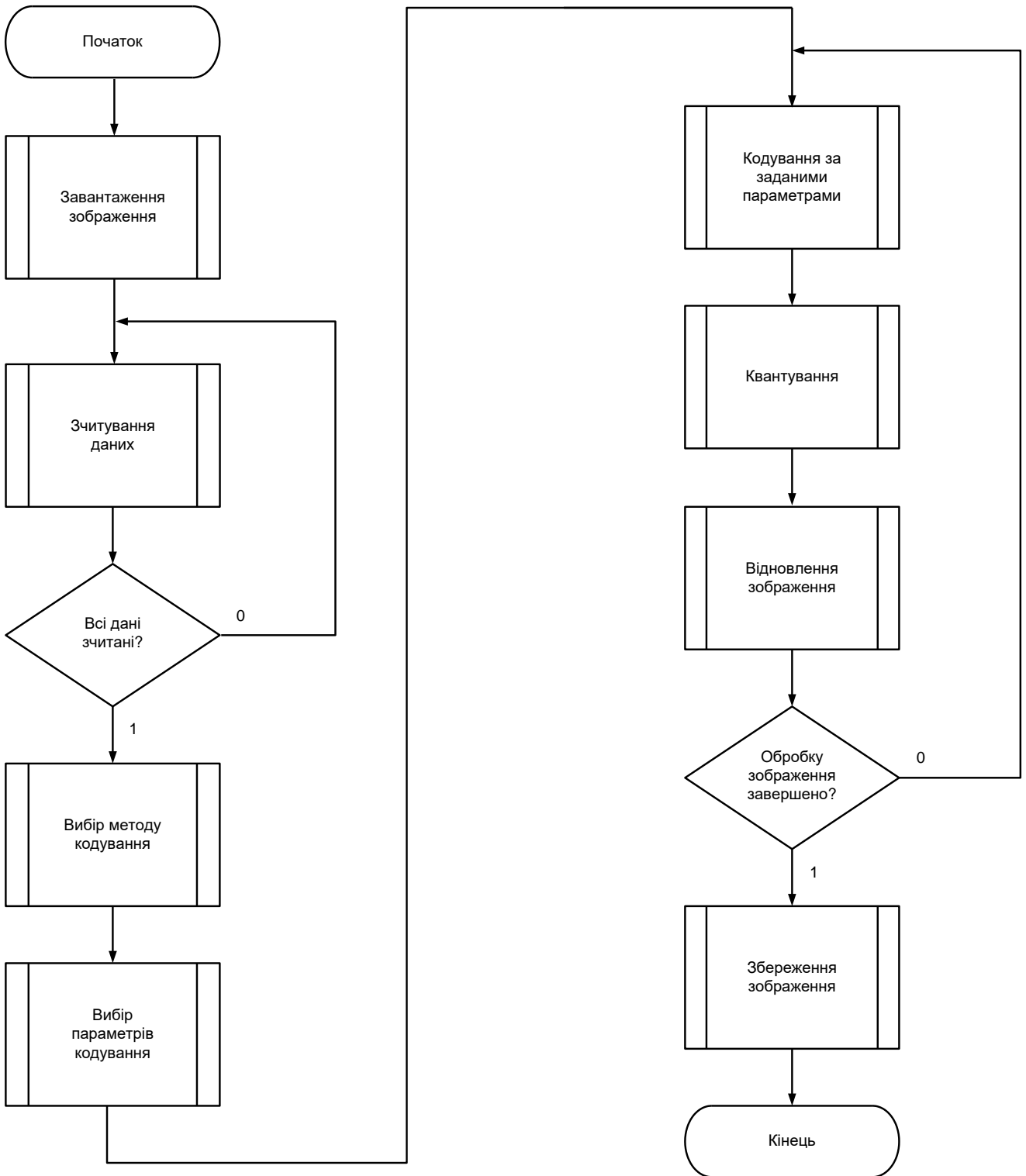


Рисунок Б.3 – Алгоритм стиснення зображень

## Експериментальні дослідження

Таблиця Б.1 – Бітові витрати на піксель зображень з рис. 4

|               |       |       |       |       |       |      |
|---------------|-------|-------|-------|-------|-------|------|
| Якість VIF    | 0.11  | 0.21  | 0.31  | 0.46  | 0.68  | 0.99 |
| Кодек         |       |       |       |       |       |      |
| Baseline JPEG | 0.159 | 0.218 | 0.349 | 0.541 | 0.939 | 3.46 |
| JPEG-MOD      | 0.132 | 0.197 | 0.332 | 0.529 | 0.935 | 3.30 |

Таблиця Б.2 – Порівняння розмірів файлів

| Ступінь стиснення | Алгоритм стиснення |          |
|-------------------|--------------------|----------|
|                   | JPEG               | JPEG-MOD |
| 50%               | 75 KB              | 66 KB    |
| 80%               | 68 KB              | 60 KB    |
| 90%               | 57 KB              | 55 KB    |

Таблиця Б.3 – Значення критеріїв якості

|          |                        | Ступінь стиснення |         |          |          |         |         |         |         |          |
|----------|------------------------|-------------------|---------|----------|----------|---------|---------|---------|---------|----------|
|          |                        | 10%               | 20%     | 30%      | 40%      | 50%     | 60%     | 70%     | 80%     | 90%      |
| JPEG     | Середньокв. відхилення | 1.0111            | 1.0361  | 1.0949   | 1.1776   | 1.3151  | 1.5413  | 2.0153  | 3.5573  | 33.0291  |
|          | PSNR                   | 0.1302            | 0.3417  | 0.8212   | 1.4537   | 2.4128  | 3.7918  | 6.1209  | 11.0564 | 30.4119  |
| JPEG-MOD | Середньокв. відхилення | 2.5175            | 7.2188  | 934.9723 | 502.9834 | 1823.3  | 8434    | 5525.3  | 37415   | 16788000 |
|          | PSNR                   | 8.0535            | 17.2033 | 59.45    | 54.0651  | 65.2509 | 78.5546 | 94.8811 | 91.495  | 164.5337 |

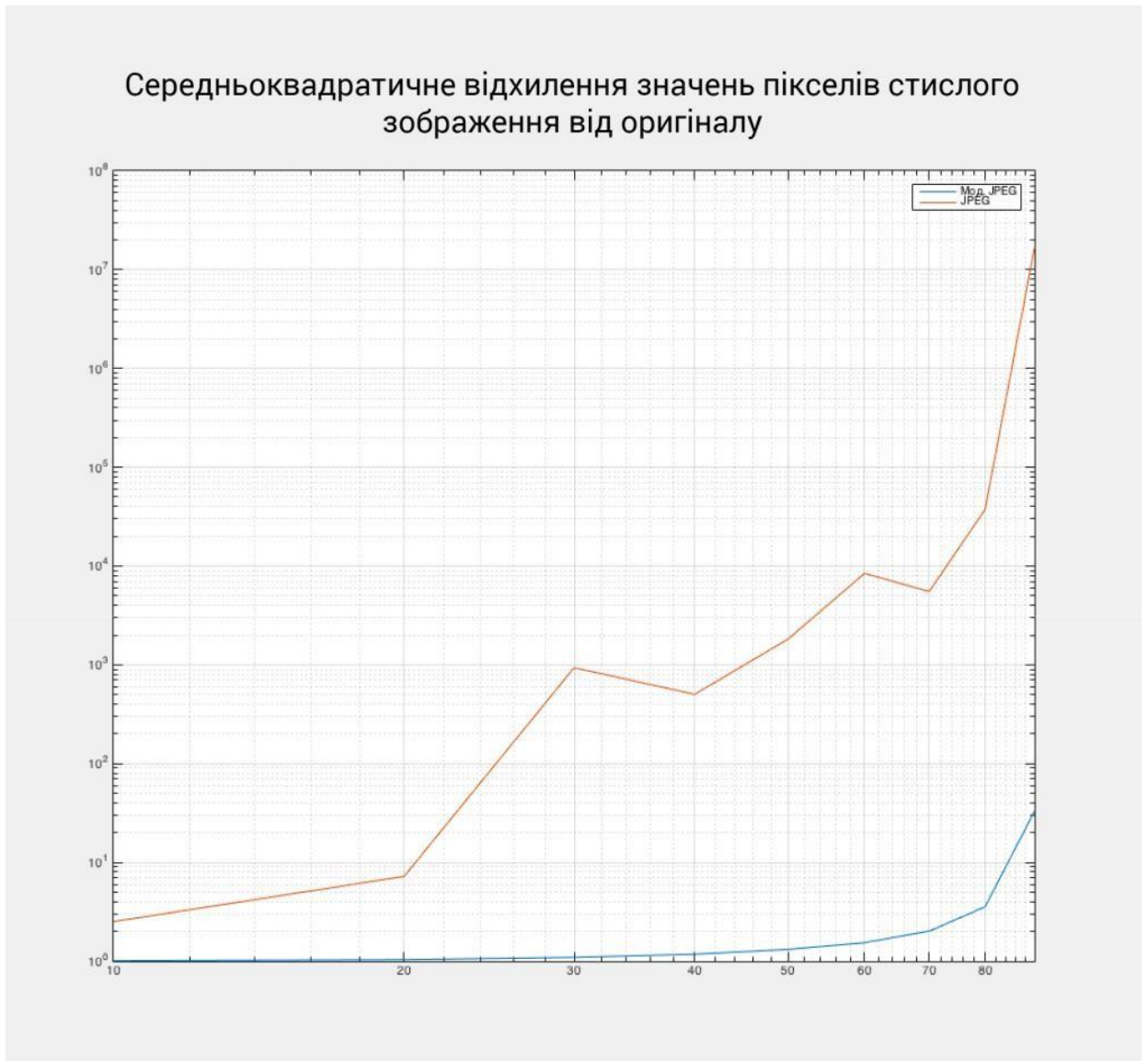


Рисунок Б.4 - Середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу

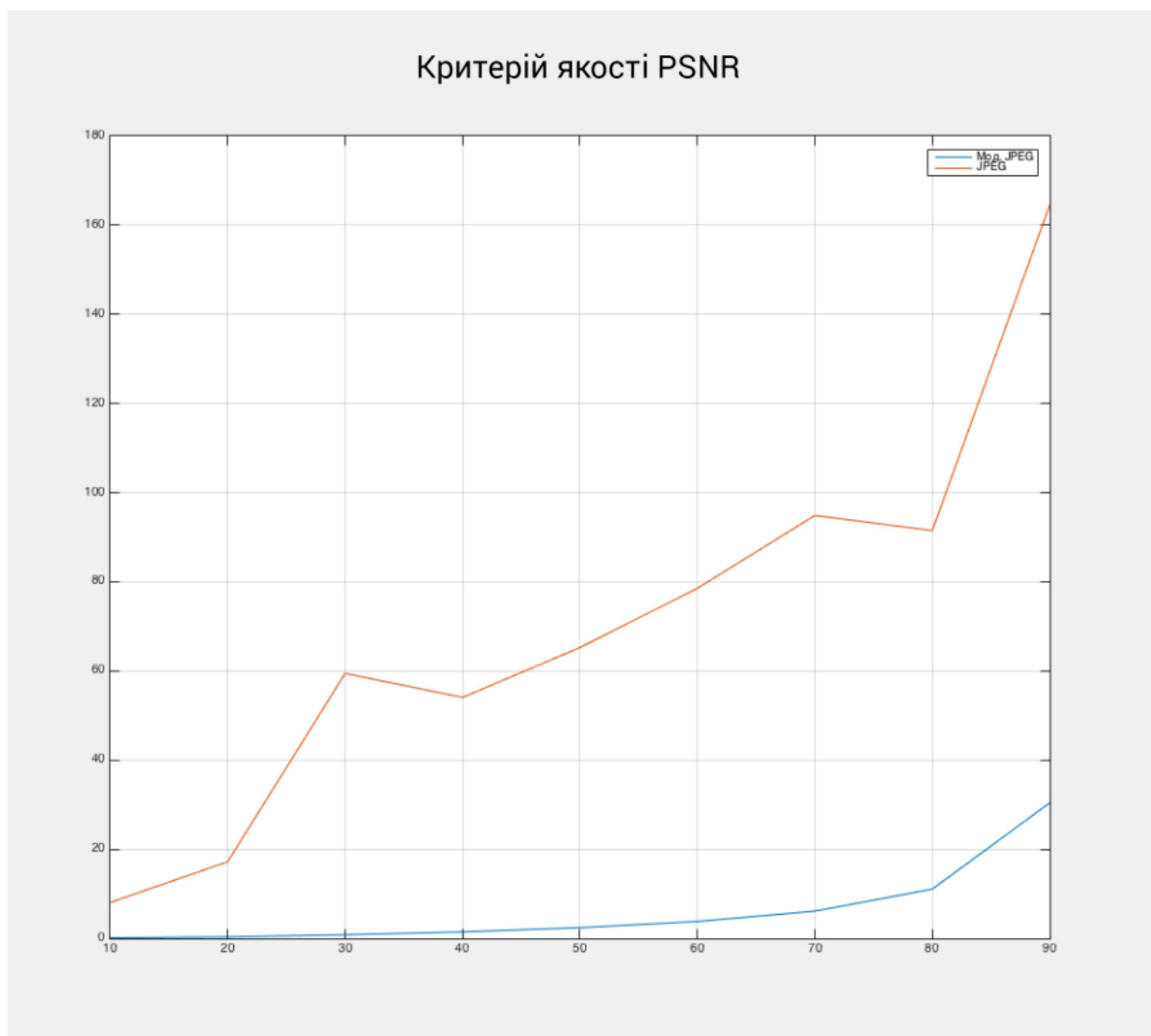


Рисунок Б.5 - Критерій якості PSNR

## ДОДАТОК В (ОБОВ'ЯЗКОВИЙ)

### Лістинг реалізації модифікованого алгоритму стиснення JPEG

```

clc;
clear all;
I = imread('cameraman.tif');
I1=I;
[row coln P]= size(I);
I= double(I);
%-----
% Subtracting each image pixel value by 128 %-----
%-----

I = I - (128*ones(256));
quality = input('What quality of compression you require - ');
%-----
% Quality Matrix Formulation
%-----

Q50 = [ 16 11 10 16 24 40 51 61;
        12 12 14 19 26 58 60 55;
        14 13 16 24 40 57 69 56;
        14 17 22 29 51 87 80 62;
        18 22 37 56 68 109 103 77;
        24 35 55 64 81 104 113 92;
        49 64 78 87 103 121 120 101;
        72 92 95 98 112 100 103 99];

if quality > 50
    QX = round(Q50.*(ones(8)*((100-quality)/50)));
    QX = uint8(QX);
elseif quality < 50
    QX = round(Q50.*(ones(8)*(50/quality)));
    QX = uint8(QX);
elseif quality == 50

```

```

    QX = Q50;

end

%-----
% Jpeg Compression
%-----

restored = zeros(row,coln);

QX = double(QX);

%-----
% Jpeg Encoding
%-----

X=I;

iter = 2; % number of wavelet iterations

cf = 1-(quality/100);

M = row;
N = coln;

% Haar basis constants

avg = 1/sqrt(2)*[1;1];
diff = 1/sqrt(2)*[1;-1];

%-----
% Compute the wavelet coefficients
%-----

tic

    WX=X;

    for i=1:iter

        %         if strcmp(type,'Haar')
        % Generate the M/(2^(i-1)) mother and father Haar wavelets mother_m =
        [avg;zeros(M/(2^(i-1))-2,1)];

        father_m = [diff;zeros(M/(2^(i-1))-2,1)];

        % Generate the N/(2^(i-1)) mother and father Haar wavelets mother_n =
        [avg;zeros(N/(2^(i-1))-2,1)];

        father_n = [diff;zeros(N/(2^(i-1))-2,1)];

        % Compute wavelet transform of upper left quadrant
        fft_mother_n_tilde = fft([mother_n(1);mother_n(end:-1:2)]');

        fft_father_n_tilde = fft([father_n(1);father_n(end:-1:2)]');

```

```

fft_mother_m_tilde = fft([mother_m(1);mother_m(end:-1:2)]);
fft_father_m_tilde = fft([father_m(1);father_m(end:-1:2)]); for p = 1:P

    for j = 1:M/(2^(i-1))

        lowfreq = ifft(fft_mother_n_tilde.*fft(WX(j,1:N/(2^(i-1)),p))); detail =
        ifft(fft_father_n_tilde.*fft(WX(j,1:N/(2^(i-1)),p))); WX(j,1:N/(2^(i-1)),p) =
        [lowfreq(1:2:end) detail(1:2:end)];

    end

    for j = 1:N/(2^(i-1))

        lowfreq = ifft(fft_mother_m_tilde.*fft(WX(1:M/(2^(i-1)),j,p))); detail =
        ifft(fft_father_m_tilde.*fft(WX(1:M/(2^(i-1)),j,p))); WX(1:M/(2^(i-1)),j,p) =
        [lowfreq(1:2:end);detail(1:2:end)];

    end

end

end

end

ForwardTransformTime = toc %#ok

%-----
%-----
% Compress image by zeroing out cf% of the wavelet coefficients %-----
-----

WXc = WX;
if cf > 0

    % Zero out smallest cf% wavelet coefficients in absolute value coeffs =
    zeros(1,M*N*P);

    for p = 1:P
        for i =
            1:M

                coeffs(M*N*(p-1) + ((i-1)*N+1:N*i)) = abs(WX(i,.,p)); end

    end

    sortedcoeffs = sort(coeffs);
    thresh = sortedcoeffs(ceil(cf*M*N*P)); indices = find((abs(WX) <= thresh));
    WXc(indices(randperm(length(indices),floor(cf*M*N*P)))) = 0;

```



```

elseif cf == -1

    % Keep the low frequency approximation and zero out everything else WXc =
    zeros(M,N,P);

    for p = 1:P
        WXc((M-M/(2^iter)+1):end,(N-N/(2^iter)+1):end,p) = WX((M-
M/(2^iter)+1):end,(N-N/(2^iter)+1):end,p);

    end

end

%-----

in = WXc;

% Z-scanning      initializing the variables
%-----

h = 1;
v = 1;
vmin = 1;
hmin = 1;
vmax = size(in, 1);
hmax = size(in, 2);
i = 1;
output = zeros(1, vmax * hmax);
%-----

while ((v <= vmax) & (h <= hmax))

    if (mod(h + v, 2) == 0)                % going up
        if (v == vmin)
            output(i) = in(v, h);          % if we got to the first line
            if (h == hmax)
                v = v + 1;
            else
                h = h + 1;
            end;
            i = i + 1;
        elseif ((h == hmax) & (v < vmax)) % if we got to the last column

```

```

        output(i) = in(v, h);
v = v + 1;
i = i + 1;
        elseif ((v > vmin) & (h < hmax)) % all other cases
            output(i) = in(v, h);
v = v - 1;
h = h + 1;
i = i + 1;
        end;
% going down
        if ((v == vmax) & (h <= hmax))          % if we got to the last line
output(i) = in(v, h);

            h = h + 1;
            i = i + 1;
        elseif (h == hmin)          % if we got to the first column
output(i) = in(v, h);

            if (v == vmax)
                h = h + 1;
            else
                v = v + 1;
            end;
i = i + 1;

        elseif ((v < vmax) & (h > hmin)) % all other cases

            output(i) = in(v, h);
            v = v + 1;
            h = h - 1;
            i = i + 1;
        end;

        end;

        if ((v == vmax) & (h == hmax))          % bottom right element
output(i) = in(v, h);

            break

```

```

end;

end;

output
%-----
% Reconstruct original image from compressed wavelet coefficients %-----
%-----

tic

Xhat =
WXC; for
i=iter:-1:1

    % Generate the M/(2^(i-1)) mother and father Haar wavelets mother_m =
    [avg;zeros(M/(2^(i-1))-2,1)];

    father_m = [diff;zeros(M/(2^(i-1))-2,1)];

    % Generate the N/(2^(i-1)) mother and father Haar wavelets mother_n =
    [avg;zeros(N/(2^(i-1))-2,1)];

    father_n = [diff;zeros(N/(2^(i-1))-2,1)]

    % Compute inverse wavelet transform of upper left quadrant fft_mother_m
    = fft(mother_m);

    fft_father_m = fft(father_m);
    fft_mother_n = fft(mother_n);
    fft_father_n = fft(father_n); for p
    = 1:P

        for j = 1:N/(2^(i-1))

            up1 = upsample(Xhat(1:M/(2^i),j,p),2);
            up2 = upsample(Xhat(M/(2^i)+1:M/(2^(i-1)),j,p),2); Xhat(1:M/(2^(i-1)),j,p) = ifft(fft_mother_m.*fft(up1(:)) +

ifft(fft_father_m.*fft(up2(:)));

            end

        for j = 1:M/(2^(i-1))

            up1 = upsample(Xhat(j,1:N/(2^i),p),2);

            up2 = upsample(Xhat(j,N/(2^i)+1:N/(2^(i-1)),p),2); Xhat(j,1:N/(2^(i-1)),p) = (ifft(fft_mother_n.*fft(up1(:)) +

```

```

ifft(fft_father_n.*fft(up2(:))));
        end
    end
end

InverseTransformTime = toc %#ok

I2 = Xhat;
% -----
% Conversion of Image Matrix to Intensity image %-----
-----

K=mat2gray(I2);
%-----
% Formulation of forward DCT Matrix and inverse DCT matrix %-----
-----

DCT_matrix8 = dct(eye(8));
iDCT_matrix8 = DCT_matrix8';          %inv(DCT_matrix8);
%-----

% Jpeg Compression
%-----

    dct_restored = zeros(row,coln);

QX = double(QX);
%-----

% Jpeg Encoding
%-----
%-----

% Forward Discret Cosine Transform
%-----

for i1=[1:8:row]
    for i2=[1:8:coln]
        zBLOCK=I(i1:i1+7,i2:i2+7);
        win1_dct=DCT_matrix8*zBLOCK*iDCT_matrix8;
        dct_domain(i1:i1+7,i2:i2+7)=win1_dct;

    end
end
end

```

```

%-----
% Quantization of the DCT coefficients
%-----
for i1=[1:8:row]
    for i2=[1:8:coln]
        win1_dct = dct_domain(i1:i1+7,i2:i2+7);
        win2_dct=round(win1_dct./QX);
        dct_quantized(i1:i1+7,i2:i2+7)=win2_dct;
    end
end
%-----
% Jpeg Decoding
%-----
% Dequantization of DCT Coefficients
%-----
for i1=[1:8:row]
    for i2=[1:8:coln]
        win2_dct = dct_quantized(i1:i1+7,i2:i2+7);
        win3_dct = win2_dct.*QX;
        dct_dequantized(i1:i1+7,i2:i2+7) = win3_dct;
    end
end
%-----
% Inverse DISCRETE COSINE TRANSFORM %-----
%-----
for i1=[1:8:row]
    for i2=[1:8:coln]
        win3_dct = dct_dequantized(i1:i1+7,i2:i2+7);
        win4_dct=iDCT_matrix8*win3_dct*DCT_matrix8;
        dct_restored(i1:i1+7,i2:i2+7)=win4_dct;
    end
end
I2_dct=dct_restored;

```

```
% -----  
% Conversion of Image Matrix to Intensity image %-----  
-----  
K_dct=mat2gray(I2_dct);  
%-----  
%Display of Results  
%-----  
K1 = single(K);  
K1_dct = single(K_dct);  
saveas(gcf,'jpg.jpg')  
figure(2);imshow(I1);  
saveas(gcf,'orig.jpg')  
figure(3);imshow(K1);  
saveas(gcf,['wavelet-',num2str(quality),'.jpg'])  
figure(4);imshow(K1_dct);  
saveas(gcf,['dct-',num2str(quality),'.jpg'])
```

ДОДАТОК Г  
(ОБОВ'ЯЗКОВИЙ)

**ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ  
(КВАЛІФІКАЦІЙНОЇ) РОБОТИ**

Назва роботи: *Модифікація алгоритму стиснення зображень JPEG на основі вейвлет-перетворення.*

Тип роботи: кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше – зазначити)

Підрозділ: кафедра АІТ, ФІТА, ІСТ-21м

(кафедра, факультет, навчальна група)

Науковий керівник: Бісікало О.В., проф. каф. АІТ

(прізвище, ініціали, посада)

Показники звіту подібності

| <i>Plagiat.pl (StrikePlagiarism)</i> |   | <i>Unicheck</i> |       |
|--------------------------------------|---|-----------------|-------|
| КП1                                  | - | Оригінальність  | 92.7% |
| КП2                                  | - |                 |       |
| Тривога/Білі знаки                   | / | Схожість        | 7.3%  |

Аналіз звіту подібності (відмітити потрібне)

**X Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор \_\_\_\_\_ Конопльов Е.Ю.  
(підпис) (прізвище, ініціали)

Опис прийнятого рішення:

Допустити до захисту

Особа, відповідальна за перевірку \_\_\_\_\_ Маслій Р.В.  
(підпис) (прізвище, ініціали)