

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«На правах рукопису»

УДК 004.93

«До захисту допущено»

Завідувач кафедри

_____ О.Р. Чертов

“ ___ ” _____ 2015 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 8.04030101 «Прикладна математика»

на тему: Ущільнення та відновлення цифрових зображень за допомогою штучних нейронних мереж

Виконав: студент 2 курсу, групи КМ-31м

Превір Микола Всеволодович

_____ (підпис)

Науковий керівник

доцент, канд. техн. наук, доцент
Олефір О. С.

_____ (підпис)

Консультант із
нормоконтролю

старший викладач Мальчиков В. В.

_____ (підпис)

Рецензент

доцент, канд. техн. наук, доцент
Корнійчук В. І.

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____ (підпис)

Київ – 2015 року

Національний технічний університет України
«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – другий (магістерський)

Спеціальність 8.04030101 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.Р. Чертов

«__» _____ 20__ р.



ЗАВДАННЯ
на магістерську дисертацію студенту
Превіру Миколі Всеволодовичу

1. Тема дисертації «Ущільнення та відновлення цифрових зображень за допомогою штучних нейронних мереж»,
науковий керівник дисертації Олєфір Олєксандр Степанович, канд. техн. наук,
доцент,
затверджені наказом по університету від «20» березня 2015р. №785-С.
2. Термін подання студентом дисертації «18» червня 2015р.
3. Об'єкт дослідження: штучні нейронні мережі типу перцептрон та мережі Хопфілда, стиснення даних з втратами та без, існуючі алгоритми квантування та стиснення зображень.
4. Предмет дослідження: методи стиснення зображень, теорія штучних нейронних мереж, формати цифрового представлення зображень.

5. Перелік завдань, які потрібно розробити:

- Проаналізувати підходи до стиснення цифрових зображень;
- Провести порівняльний аналіз різних типів штучних нейронних мереж;
- Обрати архітектуру нейронної мережі, яка є оптимальною при застосуванні в даній предметній області;
- Розробити алгоритм стиснення зображень на основі обраної архітектури нейронної мережі;
- Програмно реалізувати розроблений алгоритм;
- Провести порівняльний аналіз отриманих результатів з уже існуючими рішеннями;

6. Орієнтовний перелік ілюстративного матеріалу :

- Блок-схема роботи алгоритму
- Тестова вибірка зображень
- Стиснуті за допомогою розробленого алгоритму зображення з тестової вибірки
- Графіки, що підтверджують ефективність роботи алгоритму
- Слайди презентації з доповіді
- Зображення архітектур нейронних мереж

7. Орієнтовний перелік публікацій

- VII наукова конференція магістрантів та аспірантів “Прикладна математика та комп’ютинг - ПМК 2015”
- Міжнародна наукова конференція імені Т. А. Таран “Інтелектуальний аналіз інформації - IAI 2015”

8. Дата видачі завдання «25» жовтня 2013 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вибір напрямку дослідження та узгодження тематики МД з керівником	15 вересня – 10 листопада 2013	
2	Вивчення літератури за тематикою дисертації та збір даних	10 листопада 2013 – 8 січня 2014	
3	Формулювання постановки задачі	8 січня – 1 березня 2014	
4	Аналіз методів стиснення зображень	1 березня – 1 квітня 2014	
5	Дослідження нейромережевого підходу до стиснення даних	1 квітня – 15 травня 2014	
6	Дослідження типів нейронних мереж	15 травня – 25 червня 2014	
7	Вибір нейронної мережі для застосування для стиснення зображень та його обґрунтування	25 червня – 25 жовтня 2014	
8	Розробка алгоритму стиснення зображень за допомогою штучних нейронних мереж	25 жовтня 2014 – 1 лютого 2015	
9	Програмна реалізація	1 лютого – 1 квітня 2015	
10	Випробування розробленого алгоритму та збір статистичних даних	1 квітня – 20 квітня 2015	
11	Оформлення поточних досягнень для публікації на конференціях	20 квітня – 1 червня 2015	
12	Оформлення пояснювальної записки магістерської дисертації	1 червня – 18 червня 2015	

Студент _____

М.В. Превір

Науковий керівник дисертації _____

О.С. Олефір

РЕФЕРАТ

Магістерська дисертація складається з вступу, 7 розділів та висновків, і налічує 67 сторінок. Містить 41 ілюстративний матеріал, 2 таблиці, 2 додатки та посилається на 19 літературних джерел.

Дана магістерська дисертація присвячена проблемі стиснення цифрових зображень, та дослідженню застосування штучних нейронних мереж в цьому контексті.

В рамках дисертації на основі алгоритму Block Truncation Coding запропоновано новий алгоритм стиснення зображень з втратами, який використовує мережі Хопфілда. Алгоритм має ряд переваг за кількісними показниками якості відновленого зображення, при цьому зберігаючи рівень стиснення BTC. В роботі наведені приклади роботи, реалізованого на MATLAB алгоритму, та проведено аналіз отриманих результатів.

Результати досліджень оприлюднено на наступних конференціях: міжнародна наукова конференція імені Т. А. Таран “Інтелектуальний аналіз інформації - ІАІ 2015” (НТУУ “КПІ”), VII наукова конференція магістрантів та аспірантів “Прикладна математика та комп’ютинг - ПМК 2015” (НТУУ “КПІ”).

Ключові слова: стиснення зображень, штучні нейронні мережі, нейромережеве стиснення даних, мережі Хопфілда, перцептрон, стиснення з втратами, Block Truncation Coding, квантування зображень.

ABSTRACT

This master's thesis consists of introduction, 7 chapters and conclusions. Number of pages – 67. It includes 41 illustrations, 2 tables, 2 appendixes and references 19 literature sources.

This thesis is dedicated to the problem of digital images compression and the research of artificial neural networks application within this context.

In this thesis the new lossy image compression algorithm which uses Hopfield networks and based on Block Truncation Coding algorithm was designed. Proposed algorithm has advantages in image quality and error metric values if compared to BTC, while preserving the same compression ratio. In this paper the results of work of the algorithm implemented in MATLAB are presented. The comparative analysis of the results is fulfilled.

The results of this research were also presented on the following conferences: “Intellectual Information Analysis 2015” (NTUU “KPI”), “Applied mathematics and computing 2015” (NTUU “KPI”).

Keywords: image compression, artificial neural networks, compression neural networks, Hopfield network, perceptron, lossy data compression, Block Truncation Coding, Image quantization.

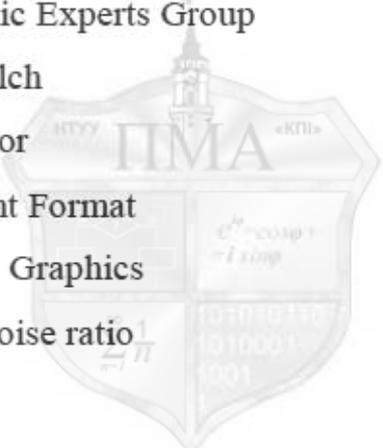
ЗМІСТ

СПИСОК ТЕРМІНІВ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1 ПОСТАНОВКА ЗАДАЧІ.....	12
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	14
2.1 Огляд існуючих рішень	14
2.2 Алгоритми стиснення зображень без втрат.....	15
2.3 JPEG.....	17
2.4 Фрактальне стиснення	18
2.5 Block truncation coding.....	19
2.6 Висновки	20
3 ВИБІР НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СТИСНЕННЯ ЗОБРАЖЕНЬ	22
3.1 Основні поняття	22
3.2 Bottle-neck мережа	23
3.3 Мережі Кохонена	24
3.4 Мережі Хопфілда	25
3.5 Висновки	29
4 БАЗОВИЙ МЕТОД СТИСНЕННЯ ЗОБРАЖЕНЬ	30
4.1 Огляд методу Block Truncation Coding.....	30
4.2 Базовий алгоритм BTC	31
5 МЕРЕЖІ ХОПФІЛДА.....	35
5.1 Базові характеристики	35
5.2 Застосування мережі Хопфілда для стиснення зображень.....	36

5.3	Алгоритм стиснення зображення.....	39
6	ВИПРОБУВАННЯ РЕАЛІЗОВАНОГО МЕТОДУ	41
6.1	Критерії оцінки ефективності стиснення зображень.....	41
6.1.1	Рівень стиснення.....	41
6.1.2	Середньоквадратична похибка (MSE).....	42
6.1.3	Співвідношення сигналу до шуму (SNR).....	42
6.1.4	Максимальне співвідношення сигналу до шуму (PSNR).....	43
6.2	Опис розроблених програмних засобів	44
6.3	Тестова вибірка зображень.....	46
6.4	Стиснення за допомогою базового ВТС.....	48
6.5	Стиснення за допомогою мереж Хопфілда.....	53
7	АНАЛІЗ РЕЗУЛЬТАТІВ.....	58
	ВИСНОВКИ	63
	СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	66
	Додаток А. Фрагменти початкового коду в MATLAB.....	68
	Додаток Б. Ілюстративний матеріал.....	72

СПИСОК ТЕРМІНІВ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

В.В. – випадкова величина
ООП – об’єктно-орієнтовне програмування
BTC – Block Truncation Coding
BTCHN – Block Truncation Coding – Hopfield Network
DCT – discrete cosine transform
GIF – Graphics Interchange Format
JPEG – Joint Photographic Experts Group
LZW – Lempel–Ziv–Welch
MSE – mean-squared error
PDF – Portable Document Format
PNG – Portable Network Graphics
PSNR – peak signal-to-noise ratio
RGB – red, green, blue
RLE – run-length encoding
SNR – signal-to-noise ratio
TGA – Truevision Graphics Adapter
TIFF – Tagged Image File Format



ВСТУП

Беззаперечним є той факт, що стиснення зображень відіграє все важливішу роль в сучасному представленні цифрових даних. Це спричинено як і класичною проблемою ущільнення даних при розміщенні на фізичних носіях, так і розвитком телекомунікаційних технологій, а саме мобільних мереж, в яких часто існують значні обмеження на об'єм та швидкість передачі даних. Тому постає проблема ефективнішого використання трафіку, значна частина якого складається з цифрових зображень.

Існує безліч форматів представлення зображень, в яких використовуються як алгоритми стиснення з втратами, так і без. Поширені зараз методи стиснення з втратами, такі як JPEG та GIF є застарілими, і часто не можуть забезпечити потреб користувачів.

Саме тому виникає потреба пошуку нових методів компресії, які виконують більш інтелектуальний аналіз даних, та є ефективнішими в тих чи інших характеристиках стиснення. За таких умов перспективним виглядає нейромеревий підхід до стиснення даних.

З моменту появи штучних нейронних мереж було здійснено багато досліджень та присвячено багато робіт, що в результаті призвело до їхнього широкого застосування в різноманітних областях, таких як управління процесами, фінансове прогнозування з розвитком обчислювальної техніки з'явилась можливість застосування даного інструменту для повсякденних задач на персональних комп'ютерах та мобільних пристроях.

Дана дисертація присвячена дослідженню штучних нейронних мереж як інструменту для стиснення зображень, в ході якого буде запропоновано та

реалізовано метод стиснення зображень, та зібрано практичні результати.вання, обробка сигналів, розпізнавання образів.



1 ПОСТАНОВКА ЗАДАЧІ

Метою даної роботи є дослідження штучних нейронних мереж та розробка методу стиснення зображень за допомогою нейронних мереж, його програмна реалізація та порівняльний аналіз результатів з існуючими рішеннями.

Предмет дослідження: методи стиснення зображень, теорія штучних нейронних мереж, формати цифрового представлення зображень.

Об'єкт дослідження: штучні нейронні мережі типу перцептрон та мережі Хопфілда, стиснення даних з втратами та без, існуючі алгоритми квантування та стиснення зображень.

В ході виконання роботи потрібно:

- Проаналізувати підходи до стиснення цифрових зображень;
- Провести порівняльний аналіз різних типів штучних нейронних мереж;
- Обрати архітектуру нейронної мережі, яка є оптимальною при застосуванні в даній предметній області;
- Розробити алгоритм стиснення зображень на основі обраної архітектури нейронної мережі;
- Програмно реалізувати розроблений алгоритм;
- Провести порівняльний аналіз отриманих результатів з уже існуючими рішеннями.

Стиснутий файл зображення не має перевищувати за розмірами початковий. Стиснення повинно відбуватися без суттєвих втрат якості.

Відновлення стиснутого зображення у реалізованій програмі повинно займати не більше 4 секунд.

Результати роботи програми повинні бути представлені в наглядній формі у вигляді оригінальних та стиснутих зображень, графіків, карт похибок.



2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

2.1 Огляд існуючих рішень

Всі існуючі методи стиснення можна поділити на дві великі категорії:

- Алгоритми стиснення без втрат:

Коли говорять про стиснення без втрат, то розуміють, що існує алгоритм, обернений алгоритму стиснення, який дозволяє точно відновити початкове зображення. Тобто, відновлене зображення є копією оригінального. Алгоритми даної категорії базуються на прибиранні надлишковості в збереженій інформації.

- Алгоритми стиснення з втратами:

Для алгоритму стиснення з втратами оберненого алгоритму не існує. Натомість існує алгоритм, який відновлює зображення, яке не точно співпадає з початковим (частина інформації втрачається). Ідея даної категорії алгоритмів полягає у відкиданні певної частини інформації, відсутність якої не сильно вплине на сприйняття зображення людиною.

Далі розглянемо алгоритми стиснення, які використовуються у таких розповсюджених форматах, як: TIFF, PNG, JPEG, JPEG2000, WebP, GIF, TGA.

2.2 Алгоритми стиснення зображень без втрат

- RLE (run-length encoding)

В даному методі групи елементів зображення, що повторюються замінюються на пару (кількість повторів, елемент, що повторюється).

Наприклад, рядок типу 0100 0100 0100 0100 0100 0100 0100 0100, що описує якусь групу пікселів буде замінена на запис типу 0100 x 8, і т.д. Застосовується цей тип стиснення в тих випадках, коли зображення має великі ділянки однакового кольору, цифрове подання яких ідентичне. В основному, цей тип стиснення застосовується для монохромних зображень, збережених в колірній моделі Bitmap. Також даний алгоритм застосовується, як один із проміжних етапів більш складних алгоритмів.

Застосовується в наступних форматах: TGA, PackBits, PCX, PLBM.

- LZW (Lempel-Ziv-Welch)

Даний алгоритм відноситься до словникових алгоритмів. Ідея цих алгоритмів полягає в кодуванні ланцюжків елементів початкової послідовності. В цьому кодуванні використовується спеціальний словник, який отримується на основі вихідної послідовності.

В алгоритмі LZW словник представляє собою таблицю, яка заповнюється ланцюжками кодування по мірі проходження алгоритму. Словник ініціалізується всіма одноелементними ланцюжками, тобто початковий словник містить алфавіт, в якому відбувається кодування. Кожного разу, коли зустрічається ланцюжок, ще не записаний в словник, він туди додається, при цьому виводиться код, що відповідає вже записаному в словник ланцюжку [1]. За рахунок того, що код є меншим за розмірами за сам ланцюжок, і відбувається стиснення.

Формати, в яких використовується LZW: GIF, TIFF, PDF.

- Кодування Хафмана

Даний алгоритм базується на частоті появи символів у вхідному повідомленні. У відповідність кожному з символів присвоюються коди, причому символам з більшою ймовірністю ставляться у відповідність більш короткі коди. Коди Хафмана мають властивість префіксності, тобто жоден з кодів не є префіксом іншого. Це дозволяє уникнути неоднозначності при декодуванні стиснутого повідомлення.

На основі вхідної таблиці частот будується дерево кодування Хафмана за наступним алгоритмом [1]:

1. Символи представляються у вигляді вільних вузлів, з вагою, що дорівнює частоті його появи.
2. Два символи з найменшою ймовірністю додаються до часткового дерева, шляхом створення їхнього батька.
3. У списку вільних символів поточні два символи замінюються одним, з вагою, яка дорівнює сумі ймовірностей цих двох символів.
4. Одній дузі, що виходить з батька в частковому дереві присвоюється біт 1, а іншій – 0.
5. Пункти, починаючи з 2-го повторюються доти, доки у списку вільних вузлів не залишиться один елемент.

Завершується алгоритм проходом по всім листкам дерева для побудови кодів всіх символів.

Використовується в наступних форматах: JPEG, PNG.

2.3 JPEG

На даний момент JPEG (Join Photographic Experts Group) є найпопулярнішим методом стиснення зображень з втратами. JPEG був створений як метод стиснення неперервно-тонових образів [2], хоча зараз використовується всюди, незалежно від образу.

На відміну від попередньо розглянутих методів, JPEG не є окремим алгоритмом, а натомість складається з послідовності алгоритмів та засобів, які дозволяють детально налаштувати рівень стиснення та кількість втрат.

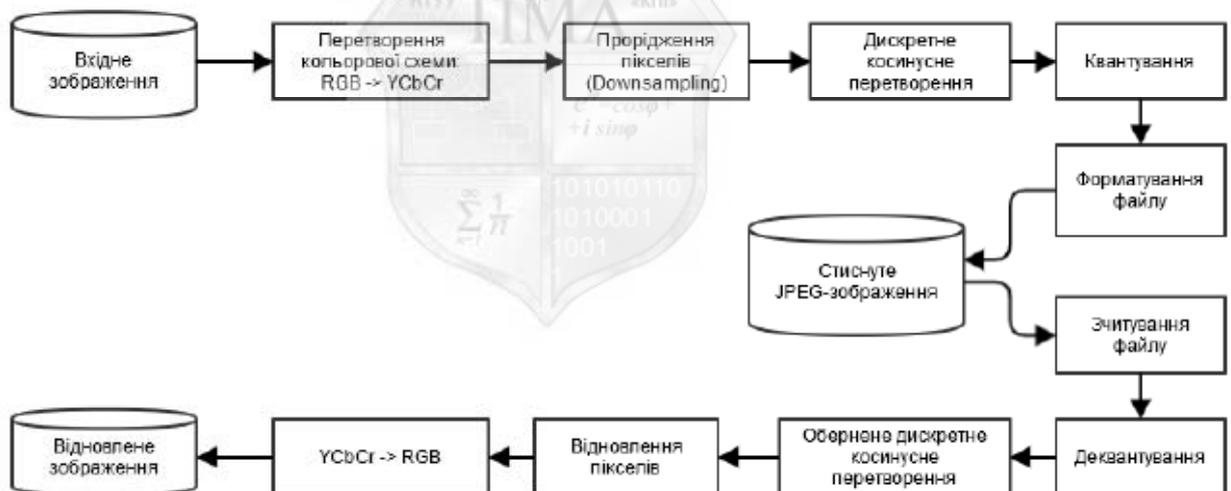


Рисунок 2.1 - Схема стиснення відновлення в JPEG.

Даний алгоритм направлений на відкидання тієї частини інформації, яка є найменш помітною для людини. Схема роботи алгоритму для стиснення та відновлення зображення представлена на рисунку 2.1. На першому кроці відбувається перехід до кольорової палітри YCbCr, де дві компоненти визначають колір, і окрема компонента його яскравість. Око людини чутливе до малих змін яскравості, але не так чутливе до невеликих розбіжностей у

кольорі. Тому дана схема дозволяє досягнути більшого стиснення в кольоровій компоненті без втрати інформації про яскравість.

Також, виходячи з тих же перцептуальних особливостей ока, прорідження пікселів відбувається шляхом об'єднання сусідніх пікселів в блоки за кольоровою компонентою. Цей крок робиться для двох компонент із трьох. Компонента яскравості залишається незмінною, тому не відбувається помітної втрати якості в зображенні [3].

2.4 Фрактальне стиснення

Фрактальне стиснення є методом стиснення з втратами і застосовується як представлення цифрового зображення у вигляді фрактальних кодів. Основа методу – пошук зразків на зображенні, що самоповторюються. Процес пошуку є ітеративним і потребує значних ресурсів, так як для пошуку зразка може знадобитись кілька мільйонів ітерацій.

Спрощена схема фрактального стиснення [4]:

- Зображення ділиться на невеликі квадратні області, що не перетинаються, які називаються ранговими блоками.
- Будується множина всіх можливих в чотири рази більших від рангових блоків, що перетинаються. Такі блоки називаються доменними. Тепер потрібно представити всі рангові блоки через невелику кількість доменних за допомогою афінних перетворень.
- Для кожного рангового блоку знаходиться найбільш подібний доменний блок, та шукається перетворення, яке найбільш наближує доменний блок до рангового. Ідеальне співпадіння для рельєфних зображень є майже неймовірним, тому даний метод є методом стиснення з втратами.

- Пара «перетворення, доменний блок», яка найбільш наблизилася до рангового блоку, ставиться йому у відповідність. При чому зберігаються лише координати доменного блоку.

Всі використані доменні блоки зберігаються без втрат окремо. Рівень стиснення сильно варіюється залежно від самоповторюваності зображення.

2.5 Block truncation coding

Block truncation coding це алгоритм квантування монохромних зображень. В результаті його виконання значення кожного з пікселів в межах одного блоку представляються за допомогою одного біту, при чому зберігаються статистичні властивості всього блоку, такі як середнє арифметичне (перший момент) та дисперсія (другий центральний момент). На рисунку 2.2 проілюстрована ідея роботи алгоритму.



Рисунок 2.2 - Схема кодування BTC.

2.6 Висновки

На сьогоднішній день існує багато методів стиснення без втрат, які можуть використовуватися для будь-яких даних. Дані методи не враховують особливості предметної області, і мають обмеження в рівні стиснення, який залежить від кількості надлишкової інформації в конкретному зображенні. Певні методи, такі як RLE та кодування Хафмана є швидкими, але вони значно поступаються за рівнем стиснення методам з втратами. Для цифрових зображень стиснення з втратами має сенс, так як сприйняття зображення людиною базується на перцептуальних особливостях, а не на кількості збереженої інформації.

Один з найефективніших та найшвидших методів стиснення з втратами – JPEG – хоч і враховує певні перцептуальні особливості зображень, спричиняє досить помітні втрати у якості зображень. Інші ж методи, які базуються на більш складному інтелектуальному аналізі інформації є ефективними тільки для певних типів зображень, та потребують значних обчислювальних ресурсів для стиснення. До таких методів можна віднести фрактальне стиснення, яке придатне лише для зображень, в яких наявні подібні образи. Ці недоліки роблять їх малопридатними для застосування для повсякденних потреб користувача.

Основна властивість нейронних мереж – виявлення взаємозв'язків між різними параметрами інформації. Це дозволить знайти та використати особливості конкретних зображень для представлення інформації більш компактно. При чому навчання нейронних мереж відбувається за допомогою простих лінійних операцій, що є значно швидшим, ніж при інших методах, таких як фрактальне стиснення.

Тому в подальшому буде розглянуто можливі способи застосування штучних нейронних мереж для стиснення зображень, а також можливість їх застосування з вже існуючими.



3 ВИБІР НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СТИСНЕННЯ ЗОБРАЖЕНЬ

3.1 Основні поняття

Штучна нейронна мережа – математична модель, принцип дії якої базується на роботі центральної нервової системи тварин.

Штучна нейронна мережа являє собою зв'язний однонаправлений граф, вузлами якого є нейрони, які мають свою функцію активації, а ребрами синапси, які мають свою синаптичну вагу. При чому граф поділений на шари так, що кожен вузол (нейрон) одного шару зв'язаний з усіма вузлами (нейронами) попереднього.

Нейронна мережа складається з множини нейронів, зв'язаних між собою синапсами з певними вагами. Характеристики нейронної мережі: вид функції активації, тип міжшарових зв'язків та алгоритм навчання.

Під навчанням мережі розуміють знаходження таких синаптичних вагів, при яких вхідний сигнал буде перетворений мережею в потрібний вихідний.

Основною властивістю нейронних мереж є визначення взаємозв'язків між різними параметрами інформації, що дозволяє нам представити її більш компактно, шляхом виявлення і відкидання надлишкової інформації (стиснення без втрат), або відкидати ту частину інформації, яка є мало важливою і її втрати ми (майже) не помітимо (стиснення з втратами).

Нейромережеве стиснення даних полягає в проектуванні і навчанні нейронної мережі таким чином, щоб отримати шар, з меншою кількістю нейронів, ніж у вхідному, такий, щоб було можливо відновити вхідний сигнал (стиснення без втрат) або близький до нього (стиснення з втратами).

3.2 Bottle-neck мережа

Дана мережа являє собою багатошаровий перцептрон з вхідним і вихідним шаром однакового розміру, в якому приховані шари є меншими, ніж вхідний та вихідний. Як і звичайний перцептрон дана мережа навчається методом зворотнього розповсюдження похибки, а функцією активації слугує сигмоїда.

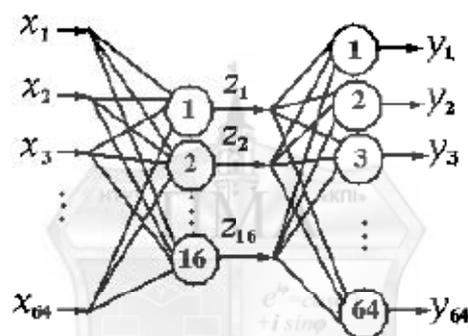


Рисунок 3.1 - Архітектура мережі Bottleneck

В мережі, зображеній на рисунку 3.1 кожен вхідний образ являє собою вектор $\vec{x} = (x_1, \dots, x_n)$, вихідний образ $\vec{y} = (y_1, \dots, y_n)$ і навчальний образ $\vec{d} = (d_1, \dots, d_n)$. В процесі навчання необхідно мінімізувати розходження значень векторів \vec{d} і \vec{y} . Формально це можна представити як задачу оптимізації наступної цільової функції [5]:

$$E(k) = \frac{1}{2} \sum_{j=1}^n (d_j(k) - y_j(k))^2 = \frac{1}{2} \sum_{j=1}^n e_j^2(k), \quad (3.1)$$

де k – номер кроку навчання.

Значення синаптичних вагів $\omega_{i,j}$ ініціалізуються випадковими малими значеннями перед першим кроком навчання. На кожному кроці відбувається послідовна адаптація вагів по мірі надходження вхідних образів. При цьому для кожної пари образів \vec{x} та \vec{d} ваги $\omega_{i,j}$ змінюються на величину $\Delta\omega_{ij}$, пропорційну антиградієнту цільової функції $E(k)$ [6]:

$$\omega_{i,j}(k + 1) - \omega_{i,j}(k) = \Delta\omega_{i,j}(k) = -\eta(k) \frac{\partial E(k)}{\partial \omega_{i,j}}, \quad (3.2)$$

де $0 < \eta < 1$ кроковий коефіцієнт, який задає «швидкість» навчання.

При застосуванні даної архітектури до стиснення даних без втрат, вектори \vec{x} та \vec{y} – ідентичні. Таким чином після завершення навчання, образ на найменшому прихованому шарі (\vec{z} на рис.1) зберігається, що і є стиснутими даними.

Для забезпеення процесу стиснення-відновлення будуються дві окремі мережі, що базуються на навченій початковій мережі. Для встановлення синаптичних вагів мережа розбивається на дві: мережу стиснення (шар x – вхідний, шар z – вихідний) та мережу відновлення (шар z – вхідний, шар y – вихідний).

3.3 Мережі Кохонена

Інший тип штучних нейронних мереж, який може бути використаний для стиснення даних та зображень - мережі Кохонена. В [7] показано, що такі мережі можуть дати кращі результати, ніж мережі зі зворотнім розповсюдженням похибки при стисненні. Мережі Кохонена використовують алгоритм навчання без вчителя.

В мережах Кохонена відсутній зворотній зв'язок, і вхідні вектори організовуються в категорії залежно від їхньої схожості одне на одного.

Для стиснення даних цим типом мереж вхідні дані розбиваються на вектори меншої розмірності для використання у якості вхідного сигналу мережі. Для кожного отриманого вхідного вектору рахується Евклідова відстань до кожного з вихідних нейронів [7]. Вхідні ваги нейрону з найменшими відстанями, разом з сусідніми нейронами змінюються, таким чином, щоб збільшити вихідний сигнал цих нейронів. Цей процес повторюється поки не буде досягнута заздалегідь визначена термінальна умова.

Після подачі на вхід мережі достатньої кількості вхідних векторів, кожен вихідний нейрон стає чутливим до групи подібних вхідних векторів, і можуть бути використані для представлення характеристик вхідних даних. Це означає, що для великої кількості вхідних векторів поданих на мережу Кохонена (вхідне оригінальне зображення), стиснутою формою будуть дані, що отримуються з вихідного шару нейронів.

Стиснуті дані потім відновлюються також за допомогою мережі Кохонена.

3.4 Мережі Хопфілда

Мережа Хопфілда – це повнозв'язна штучна нейронна мережа, що складається з одного шару нейронів, який одночасно є вхідним і вихідним шаром. Матриця вагів мережі Хопфілда є симетричною. Кожен нейрон може мати один із двох станів: 1 та -1.

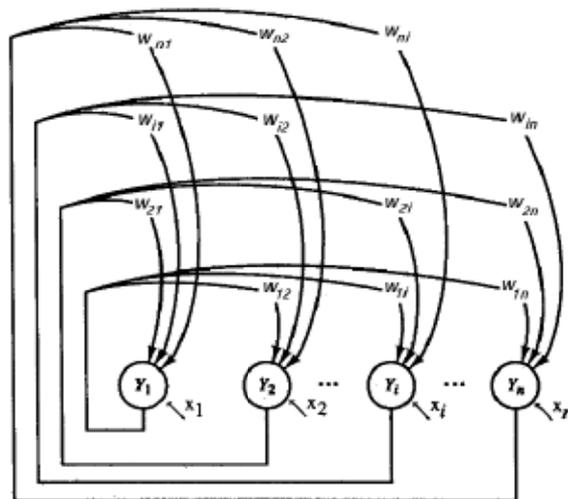


Рисунок 3.2 - Мережа Хопфілда

Якщо сума всіх вхідних сигналів, помножених на ваги відповідних зв'язків перевищує певне, заздалегідь зазначене порогове значення θ , то на виході нейрону отримаємо 1, інакше -1. Це описується наступною формулою [7]:

$$y_i = \text{sign} \left(\sum_{j=1, j \neq i}^n \omega_{ji} x_j - \theta_i \right) \quad (3.3)$$

В мережі Хопфілда існує двонаправлений зв'язок між будь-якою парою нейронів. Це легше проілюструвати за допомогою мережі з 3-х елементів на рисунку 3.3.

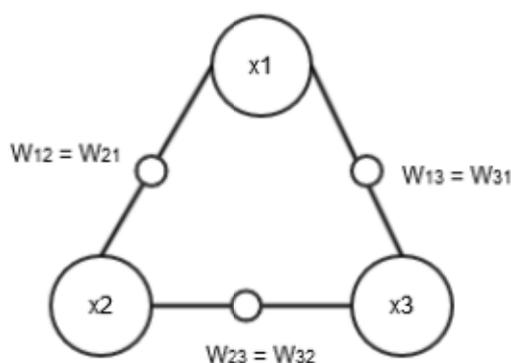


Рисунок 3.3 - Мережа Хопфілда з 3-х елементів

Мережа Хопфілда реалізує автоасоціативну пам'ять, і може відновлювати пошкоджені та зашумлені образи. Завдяки архітектурі даної мережі процес навчання мережі простий, і полягає в обрахуванні матриці вагів за один цикл [8]:

$$\omega_{ij} = \frac{1}{n} \sum_{d=1..m} x_{di} x_{dj} , \quad (3.4)$$

де m – кількість образів, що запам'ятовуються, d – номер образу, що запам'ятовується, x_{di} – i -й елемент d -го образу.

Для навченої мережі відновлення образу відбувається ітеративно [8]:

$$X_{t+1} = f(WX_t) , \quad (3.5)$$

де W – матриця вагів, X_t вектор образу на t -ій ітерації.

Ітерації відбуваються до тих пір, поки мережа не досягне стійкого стану (вихідні значення перестануть змінюватись). Мережі Хопфілда мають скалярне значення, співставлене з кожним станом мережі, яке називається енергією. Стан стійкості є локальним мінімумом функції енергії, яка визначена наступним чином [8]:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1, i \neq j}^N \omega_{ij} x_i x_j + \sum_{i=1}^N \theta_i x_i \quad (3.6)$$

При зміні стану мережі, її енергія зменшується або залишається незмінною.

Вихідне значення мережі Хопфілда, що має n нейронів можна представити як n -бітове бінарне число. Така мережа може бути в одному з 2^n станів.

Розглянемо приклад з рис. 3.3. Дана мережа може знаходитися в одному з $2^3 = 8$ станів, які позначимо через S_i , $i = 1 \dots 8$. Збережений образ, до якого сходиться мережа залежить від вхідного сигналу та матриці вагів. Нехай енергія цих станів співвідноситься наступним чином:

$$S_1 < S_2 < S_3 < S_4 < S_5 < S_6 < S_7 < S_8$$

Тоді, мережа буде сходиться до стійкого стану за наступною схемою:

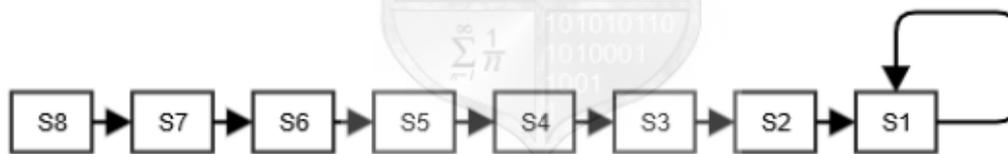


Рисунок 3.4 - Перехід станів в мережі

Як це видно з рисунку 3.4, стійким є стан S_1 , так як в ньому енергія мережі приймає своє мінімальне значення.

Роботу мережі в режимі відновлення пошкоджених даних можна застосувати для стиснення даних: з початкового образу відкидається така частина інформації, без якої відновлений матиме допустимий рівень втрат (стиснення з втратами). Також для більшої економії пам'яті, мережу можна використовувати для стиснення групи файлів. В такому випадку для кількох

файлів потрібно зберігати метадані тільки однієї мережі (кількість нейронів та матрицю вагів) [9].

Також властивості мережі Хопфілда можна використати для представлення цифрових зображень більш компактно шляхом квантування. Так як на виході у нейронів мережі Хопфілда ми отримуємо два можливих значення (кількість інформації в один біт).

3.5 Висновки

Методи стиснення, які базуються суто на штучних нейронних мережах застосовні лише для певних типів даних та мають ряд недоліків [10]. Штучні нейронні мережі – це потужний інструмент, але сам по собі не дає ефективних результатів при стисненні зображень. Тому їх варто застосовувати в комбінації з іншими методами, або як проміжний етап в алгоритмі стиснення.

Мережі Хопфілда можна використати для побудови схеми квантування на основі алгоритму ВТС.

Ідея застосування мереж Хопфілда полягає в її використанні для отримання одного із двох можливих квантованих значень пікселя в блоці, замість використання порогової функції, як це робиться в ВТС. Результат квантування в такому випадку полягає в тому, який стійкий стан був досягнутий для кожного окремого блоку зображення.

Далі буде детально розглянуто алгоритм ВТС та можливість його модифікації шляхом застосування мережі Хопфілда.

4 БАЗОВИЙ МЕТОД СТИСНЕННЯ ЗОБРАЖЕНЬ

4.1 Огляд методу Block Truncation Coding

Block Truncation Coding – це простий і швидкий алгоритм стиснення з втратами для цифрових монохромних зображень. Основна ідея BTC – здійснити квантування блоків пікселів за умови збереження їхніх статистичних характеристик (моментів), таким чином, щоб якість зображення залишилась задовільною, а розмір кінцевих даних був меншим.

Перший варіант алгоритму BTC був запропонований Делпом та Мітчелом в 1979р.

Переваги алгоритму BTC [12]:

- Простіша імплементація в порівнянні з іншими блочними методами стиснення;
- Високий рівень квантування;
- Хороша якість відновлених зображень;
- Висока швидкість роботи (в п'ять разів швидше дискретного косинусного перетворення);
- Відносно високий рівень стиснення.

В своєму початковому варіанті BTC був направлений на зберігання першого та другого моментів блоку пікселів. На даний момент існують модифікації алгоритму, які направлені на підвищення ефективності кодування базового алгоритму. Вони полягаються у комбінуванні BTC з іншими методами обробки зображень, такими як застосування медіанного фільтру та

векторне квантування [13]. В даній роботі використовуватиметься тільки базовий алгоритм ВТС. Розглянемо його більш детально у наступному пункті.

4.2 Базовий алгоритм ВТС

Загальна схема роботи алгоритму ВТС зображено на рис. 2.2. Згідно неї першим кроком є розбиття цифрового зображення на невеликі блоки розміром $n * n$. Подальше кодування відбувається для кожного блоку окремо.

Нехай $M = n \times n$, а $x(1), x(2), \dots, x(M)$ – значення пікселів з блоку оригінального зображення. Тоді середнє значення та дисперсія випадкової величини x , що є її першим та другим моментом, відповідно, визначається наступним чином:



$$\bar{x} = \frac{1}{M} \sum_{i=1}^M x(i) \quad (4.1)$$

$$\overline{x^2} = \frac{1}{M} \sum_{i=1}^M x^2(i)$$

$$\sigma^2 = \overline{x^2} - \bar{x}^2 \quad (4.2)$$

Потім будується однобітовий квантизатор. Для цього визначається порогове значення X_t , де t – номер поточного блоку, та два рівні відновлення - a і b . Нехай y_1, y_2, \dots, y_M – значення пікселів після квантування. Тоді квантизатор виглядає наступним чином:

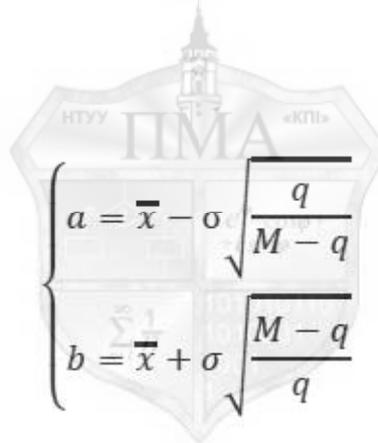
$$y_i = \begin{cases} a, & x(i) < X_t \\ b, & x(i) \geq X_t \end{cases}, \quad i = 1, 2, \dots, M$$

Якщо взяти в якості порогового значення X_t середнє значення в.в. x , то для збереження значень перших двох моментів після квантування необхідно виконання наступних умов:

$$\begin{cases} M\bar{x} = (M - q) \times a + qb \\ M\bar{x}^2 = (M - q) \times a^2 + qb^2 \end{cases}, \quad (4.3)$$

де q – число, більше за порогове значення (середнє арифметичне в.в. x).

Розв'язавши систему (4.3) відносно змінних a і b , і застосувавши (4.2) отримаємо:



$$\begin{cases} a = \bar{x} - \sigma \sqrt{\frac{q}{M - q}} \\ b = \bar{x} + \sigma \sqrt{\frac{M - q}{q}} \end{cases}$$

В результаті отримуємо бінарний квантизатор, який діє за принципом, зображеним на рисунку 4.1.

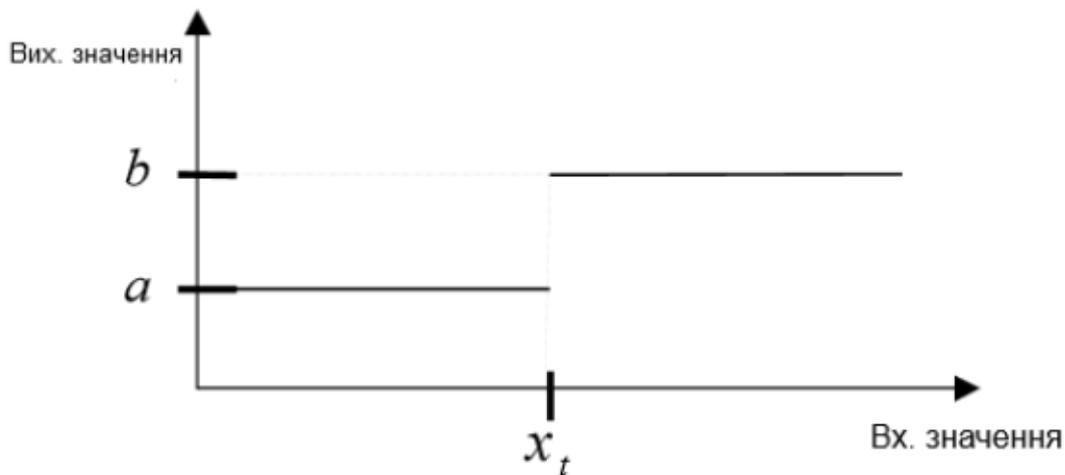


Рисунок 4.1 - Бінарний квантизатор

В 1985р. Адпікаром і Райною [14] було запропоноване покращення алгоритму. Воно полягало в збереженні статистичної інформації лише першого порядку, а саме середнього арифметичного значення пікселів блоку. Значення рівнів відновлення тоді обраховуються за наступною формулою [14]:

$$\begin{cases} a = \frac{1}{M-q} \sum_{i=1}^M x(i), & x(i) < X_t \\ b = \frac{1}{q} \sum_{i=1}^M x(i), & x(i) \geq X_t \end{cases} \quad (4.4)$$

Порогове значення та рівні відновлення квантизатору змінюються разом із статистичними характеристиками блоку. Іншими словами, кодування відбувається в межах локальної області зображення.

Після квантування блок буде представлений у вигляді асоціативної матриці $n*n$. Елементами цієї матриці пікселі, значення яких може приймати один із двох станів (тобто об'єм пам'яті, який використовується кожним із елементів дорівнює 1 біту), та визначають інтенсивність конкретного пікселя на цифровому зображенні. По суті, дана матриця є картою пікселів (bitmap).

Рівні відновлення зберігаються окремо для кожного блоку в кінцевому файлі без змін.

Процес відновлення зображення полягає у записі значень a і b у відповідні позиції кінцевого зображення згідно з бітмар кожного блоку. При цьому нулям в матриці присвоюються значення a , одиницям - b .

Перевагою алгоритму ВТС - є те, що це являється блочним алгоритмом, тому похибки в межах одного блоку ніяк не впливають на похибки в інших областях зображення при його відновленні.

Розглянемо приклад стиснення монохромного зображення за допомогою ВТС з розміром блоку 4×4 . Так як це монохромне зображення, то кожен піксель представляється 8 бітами (може приймати один з 256 відтінків сірого). Об'єм пам'яті, що використовується при такому представленні: $4 \times 4 \times 8 = 64$ біти.

Після застосування ВТС кожен піксель представляється за допомогою одного біту. Порядок розміщень бітів в блоці зображення визначений в алгоритмі і є однаковим для всіх блоків. Додатково потрібно зберегти два відновлювальних рівні, що займають 8 біт кожен (цілі невід'ємні числа). Тоді в стиснутому стані той самий блок займатиме $4 \times 4 + 8 + 8 = 32$ біти. На простому прикладі, видно, що ми отримуємо стиснення в два рази. Це справедливо, лише коли враховується тільки корисна інформація блоку (так званий розмір нетто).

На практиці значення дещо відрізняються, що пояснюється особливостями розміщення даних в файловій системі, вирівнювання сторінок пам'яті та додаткова інформація середовищ й інструментів, за допомогою яких виконується програма.

Рівень стиснення сильно варіюється в залежності від обраного розміру блоку та форми збереження рівнів відновлення. При виборі розміру блоку слід іти на компроміс між якістю та рівнем стиснення зображення.

5 МЕРЕЖІ ХОПФІЛДА

5.1 Базові характеристики

У 1982 р. Джон Хопфілд виявив, що стійкість вихідного сигналу групи нейронів можна використати для запам'ятовування вхідних образів. Аналіз ґрунтувався на мінімізації енергії мережі, поняття якої було введено Хопфілдом, та доведенням того, що робота мережі призводить до мінімізації значення енергії та приведення мережі до стійкого стану [8].

Вибіркові нейрони в мережі Хопфілда мають сильну зв'язність (відносно великі абсолютні значення відповідних вагів в матриці вагів), що забезпечує групові обчислювальні властивості (collective computational properties) мережі та їхню ефективність [15], такі як приведення мережі до конкретного стану, та виведення визначеного вихідного сигналу.

Найважливіша групова властивість нейронів в мережі Хопфілда впливає з визначення функції енергії мережі, та полягає у сходженні до стійкого стану: для заданого набору вхідних сигналів вихідне значення мережі сходиться до одного з локальних мінімумів функції енергії.

Нейрони в мережі Хопфілда можуть мати один з двох станів. Стани окремих нейронів мережі змінюються асинхронно, і кожна зміна стану призводить до зменшення абсолютного значення енергії.

Мережа Хопфілда отримує сигнал з двох джерел: зовнішній сигнал (позначимо його через I) та вихідні сигнали інших нейронів. Загальний вхідний сигнал нейрону можна поррахувати як суму цих значень:

$$H_i = I_i + \sum_{i \neq j} \omega_{ij} v_j, \quad (5.1)$$

де ω_{ij} – значення вагового коефіцієнту між i -им та j -им нейроном.

Вихідне значення нейрону визначимо наступним чином:

$$v_i = \begin{cases} 1, & \text{якщо } H_i \geq 0 \\ 0, & \text{якщо } H_i < 0 \end{cases} \quad (5.2)$$

Тоді обчислювальна енергія мережі в загальному випадку прийме наступний вигляд:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1, i \neq j}^N \omega_{ij} v_i v_j - \sum_{i=1}^N I_i v_i \quad (5.3)$$

5.2 Застосування мережі Хопфілда для стиснення зображень

На вхід мережі зображення подаються попіксельно, тобто кількість нейронів в мережі дорівнює кількості пікселів у вхідному зображенні.

На відміну від алгоритму ВТС, де для класифікації пікселів використовується середнє арифметичне в якості порогового значення, тут буде використовуватись вихідне значення мережі Хопфілда для розбиття пікселів блоку на два класи.

Блок-схема алгоритму представлена на рисунках 5.1 та 5.2.

Наступна формула визначає енергію, мінімальне значення якої відповідає ефективній класифікації пікселів [16]:

$$E = \sum_{i=1}^N \sum_{j=1}^N v_i v_j (x(i) - x(j))^2 + \sum_{i=1}^N \sum_{j=1}^N (1 - v_i)(1 - v_j) [x(i) - x(j)]^2, \quad (5.4)$$

де N – кількість пікселів в блоці;

$x(i)$, $i=1, 2, \dots, N$ – значення пікселів.

З рівняння (5.4) видно, що значення пікселів з одного класу не мають значно відрізнятись, тому що інакше мережа не буде стабільною. Тому стан мережі Хопфілда прямує до такого, при якому пікселі з близькими значеннями групуються в один клас.

Для кодування, зображення розбивається на блоки розміром $n*n$, і передається на вхід мережі Хопфілда, яка складається з $N=n*n$ нейронів. За невеликих розмірів блоку пікселі можуть бути ефективно класифіковані мережею на два класи.

Сила міжнейронних зв'язків (вагові коефіцієнти) та зовнішній вхідний сигнал в (5.4) обраховуються за наступними формулами [16]:

$$\omega_{ij} = -4(x(i) - x(j))^2$$

$$I_i = 2 \sum_{j=1}^N (x(i) - x(j))^2$$

Ітерації в нейронній мережі відбуваються доти, доки не досягнуто стабільного стану, тобто подальші ітерації не змінюють значення v_i .

Визначимо вихідні значення квантизатора, як середнє арифметичне для кожного з класів «0» та «1»:

$$\bar{X}_0 = \frac{1}{N - q} \sum_{i=1}^N (1 - v_i) x(i)$$

(5.5)

$$\bar{X}_1 = \frac{1}{q} \sum_{i=1}^N v_i x(i)$$

де q – кількість пікселів в класі $v_i = 1$.



5.3 Алгоритм стиснення зображення

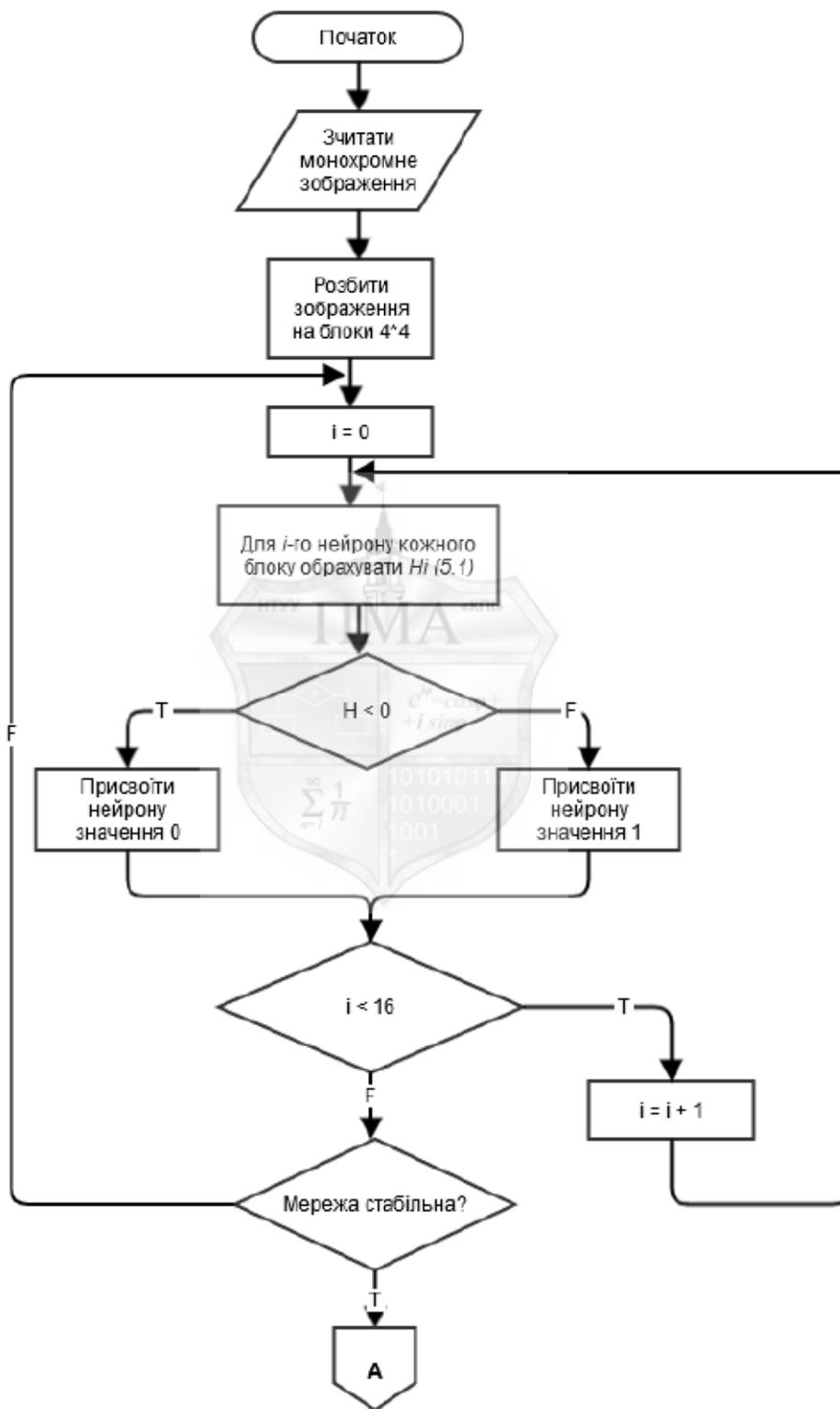


Рисунок 5.1 - Блок-схема розробленого алгоритму стиснення монохромних зображень. 1-а частина



Рисунок 5.2 - Блок-схема розробленого алгоритму стиснення монохромних зображень. 2-а частина

6 ВИПРОБУВАННЯ РЕАЛІЗОВАНОГО МЕТОДУ

6.1 Критерії оцінки ефективності стиснення зображень

Якість стиснення цифрового зображення залежить від роздільної здатності зображення. Просторова роздільна здатність зображення – це найменше число помітних різних ліній на одиницю довжини (на відміну від піксельної роздільної здатності, яка характеризується лише кількістю пікселів).

Більша просторова роздільна здатність під час відновлення означає менший рівень стиснення, кращу якість, нижчий MSE та вищий SNR.

6.1.1 Рівень стиснення

Це стандартний коефіцієнт для виміру зменшення розміру в представленні даних після застосування до них алгоритму стиснення. Якщо позначити через B - розмір даних до стиснення, A – розмір даних після застосування алгоритму стиснення, то рівень стиснення визначається наступним чином:

$$C_R = \frac{B}{A}$$

Рівень стиснення – це найпростіший критерій оцінки ефективності роботи алгоритму, який підходить до будь-якого типу даних.

6.1.2 Середньоквадратична похибка (MSE)

Даний критерій застосовується для оцінки методів стиснення зображень з втратами.

Нехай початкове зображення $f(x,y)$ було стиснено, і після відновлення отримали зображення $f'(x,y)$. Для зображення розміром $m \times n$ середньоквадратична похибка (mean square error, MSE) визначається наступним чином [17]:

$$MSE = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f(x,y) - f'(x,y))^2$$

MSE є кількісною характеристикою. Нижче значення MSE означає кращу якість відновленого зображення.

6.1.3 Співвідношення сигналу до шуму (SNR)

При обрахуванні співвідношення сигналу до шуму (signal-to-noise ratio, SNR) всі похибки у відновленому зображенні сприймаються як «шум», а оригінальне зображення – як сигнал. SNR порівнює рівень бажаного сигналу та рівень фонового шуму.

Одиниця виміру SNR – децибел. SNR рахується наступним чином [1]:

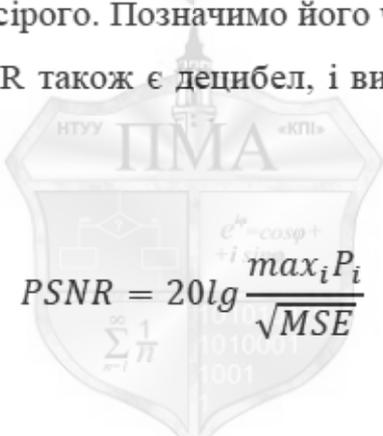
$$SNR = 10 \lg \left(\frac{\sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f^2(x,y)}{m \times n \times MSE} \right)$$

Вище значення SNR відповідає вищій якості відновленого зображення, тобто залежність є прямопропорційною.

6.1.4 Максимальне співвідношення сигналу до шуму (PSNR)

Ще одним критерієм оцінки якості стиснутого зображення є максимальне співвідношення сигналу до шуму (peak signal-to-noise ratio, PSNR). Максимальним значенням сигналу для монохромного зображення є найінтенсивніший відтінок сірого. Позначимо його через P .

Одиницею виміру PSNR також є децибел, і визначається воно наступним чином [1]:



$$PSNR = 20 \lg \frac{\max_i P_i}{\sqrt{MSE}}$$

В загальному випадку для кодування одного кольору виділяється один байт, тому значення в чисельнику дорівнюватиме 255. Для кольорових зображень використовуються лише компоненти кольору (наприклад, для кольорового простору YCbCr – компонента яскравості не береться до уваги).

Чим більша схожість між оригінальним (сигналом) та відновленим зображенням, тим більше значення PSNR.

В даній роботі будуть використані розглянуті метрики для аналізу реалізованих алгоритмів.

6.2 Опис розроблених програмних засобів

В ході виконання роботи було реалізовано базовий алгоритм BTC та його модифікація за допомогою мереж Хопфілда, описана в розділі 5. Реалізація здійснена за допомогою мови програмування та пакету прикладних програм MATLAB.

Мова програмування MATLAB націлена на вирішення складних математичних задач, тому була обрана як оптимальний інструмент для реалізації запропонованих алгоритмів. Також в MATLAB є стандартні засоби роботи з зображеннями, що дозволяють легко зчитувати і зберігати файли зображень.

Так як на меті було не створити прикладну програму, а розробити тестовий додаток для оцінки ефективності запропонованого алгоритму – програма була створена без урахувань принципів ООП, та не розрахована для подальшого використання іншими додатками. Обидва алгоритми реалізовані в межах окремих функцій, в тілі яких відбувається зчитування, стиснення, форматування та записування файлів.

Базовий алгоритм BTC реалізований в функції `btc(fileinput, fileoutput)`, де `fileinput` та `fileoutput` – назви вхідного та вихідного файлів, відповідно.

Модифікований алгоритм BTC з використанням мереж Хопфілда реалізовано в функції `bthcode (fileinput, bx, by, fileoutput)`, де

`fileinput` – назва файлу з оригінальним зображенням, що стискається;

`bx, by` – розміри блоків, на які розбиватиметься оригінальне зображення;

`fileoutput` – шлях для збереження вихідного стиснутого файлу.

Для використання цих функцій, необхідна наявність встановленого пакету прикладних програм MATLAB. Виклик цих функцій відбувається з командного рядка MATLAB.

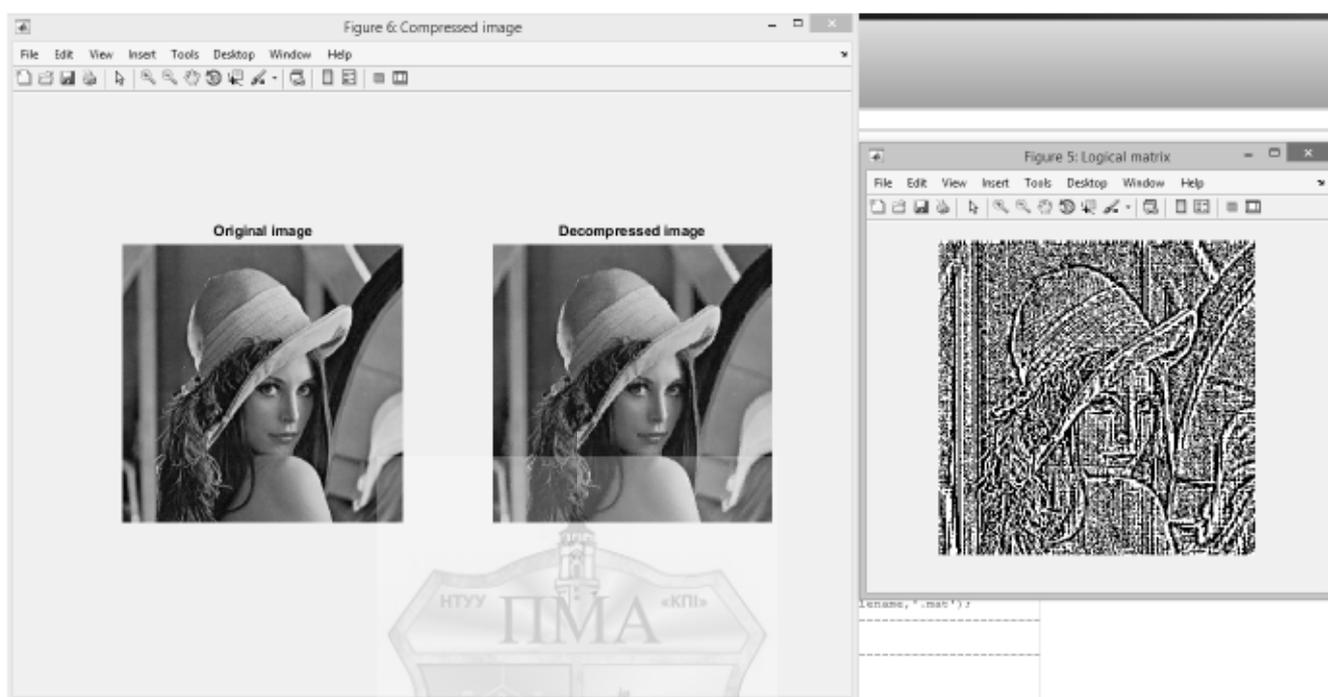


Рисунок 6.1 - Приклад роботи програми

MATLAB містить компоненти, що дозволяють працювати з більшістю сучасних форматів зображень, а також компоненти, що дозволяють швидко реалізувати графічний інтерфейс та виведення зображень (побудова фігур).

Тому, з урахуванням розглянутих характеристик, для реалізації методу було обрано середовище розробки MATLAB.

6.3 Тестова вибірка зображень

Запропонований метод буде протестовано на вибірці зображень, яка включає в себе як і класичні для сфери обробки зображень тестові об'єкти (lena.tif, cameraman.tif), так і звичайні фотографії.

Тестові зображення представлені на рисунках 6.2 – 6.6.



Рисунок 6.2 - cameraman.tif



Рисунок 6.3 - lena.tif



Рисунок 6.4 - fishingboat.tif



Рисунок 6.5 - tank.tif



Рисунок 6.6 - truck.tif

Зображення lena.tif добре підходить для випробування методів обробки зображень, так як містить високий рівень деталізації, монотонні ділянки та ділянки з плавною переменою кольорів, тому є репрезентативною [19]. sametaman.tif містить ділянки з різким контрастом кольорів, та ділянки чорого кольору (нульове або близьке до нуля значення пікселів).

6.4 Стиснення за допомогою базового BTC

Для випробування алгоритму BTC було обрано розмір блоку 4*4 пікселі. Перший блок вхідного зображення «Cameman.tif» виглядає наступним чином:

156	157	160	159
156	157	159	158
158	157	156	156
160	157	154	154

Тепер, взявши в якості порогового значення блоку його середнє арифметичне: $X = 157.125$, отримаємо наступний бітмар:

0	0	1	1
0	0	1	1
1	0	0	0
1	0	0	0

Порахуємо рівні відновлення, або, простіше кажучи, середні значення пікселів, що відповідають «0» та «1», окремо: $a = 156$, $b = 159$. Ці значення разом з бітмар змерігаються в стисненому файлі.

Після відновлення блок прийме наступний вигляд:

156	156	159	159
156	156	159	159
159	156	156	156
159	156	156	156

Розглянемо зображення «cameraman.tif» вцілому.



Рисунок 6.7 - Оригінальне зображення



Рисунок 6.8 - Bitmap

На рис. 5.2. оригінальне вхідне зображення в форматі TIFF. На рис. 5.3 – bitmap цього зображення, розбитого на блоки 4*4. Для його представлення використовуються лише «0» та «1» (чорний та білий колір, відповідно).

Після відновлення зображення виглядає наступним чином:



Рисунок 6.9 - Відновлене зображення

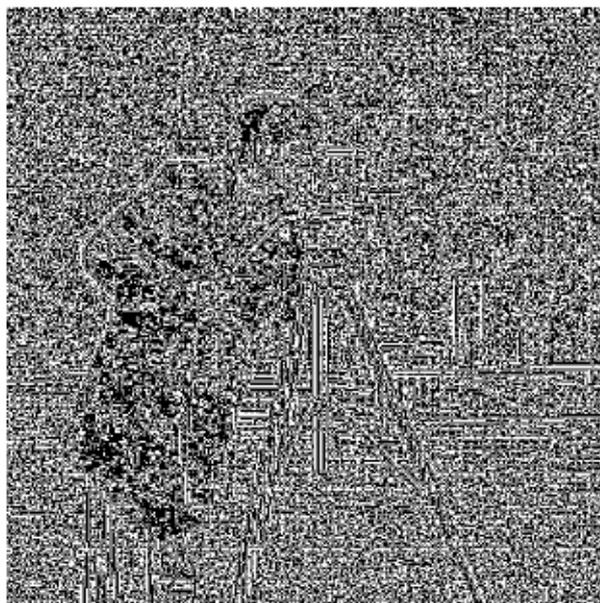


Рисунок 6.10 - Похибки

Зображення було відмасштабовано задля більшої інформативності ілюстрацій, а саме, щоб на ньому було краще видно межі блоків. Відмінності між оригінальним та відновленим зображенням легко помітні. Сусідні блоки відрізняються різким перепадом кольорів, і формують контурні артефакти. Це є одним з недоліків ВТС.

Віднявши попіксельно від оригінального зображення стиснуте – отримуємо наглядне відображення похибок (рис. 5.5). Темніший колір означає менший рівень похибок (чорний – співпадіння з оригіналом). На рис.5.5 видно, що області зображення з волоссям та пальто фотографа мають найменший рівень похибок, так як ці області є монотонними. Найбільший рівень похибок спостерігається на межі між фотографом та фоном. Це спричинено варіацією значень пікселів в межах одного блоку.

Застосуємо метод ВТС до інших зображень з тестової вибірки:

Lena.tiff



Рисунок 6.11 - Оригінальне зображення



Рисунок 6.12 - Відновлене зображення

fishingboat.tif



Рисунок 6.13 - Оригінальне зображення



Рисунок 6.14 - Відновлене зображення

tank.tiff



Рисунок 6.15 - Оригінальне
зображення



Рисунок 6.16 - Відновлене зображення

truck.tiff



Рисунок 6.17 - Оригінальне
зображення



Рисунок 6.18 - Відновлене зображення

Відмінності в якості зображення очевидні при застосуванні BTC, через появу блоків. Також було обраховано кількісні характеристики при стисненні кожного з зображень.

Табл.6.1. Кількісні характеристики зображень, стиснутих за допомогою BTC

Зображення	Розмір до стиснення	Розмір після стиснення	Коеф. стиснення	MSE	SNR	PSNR
cameraman.tif	64KB	21.276KB	0.23234	141.9874	21.0259	26.5399
Lena.tif	278KB	91.256KB	0.22825	84.5126	21.737	28.6198
fishingboat.tif	252KB	90.072KB	0.25743	101.3498	22.73	28.0726
tank.tif	256KB	94.912KB	0.27075	39.9268	26.6018	30.9535
truck.tif	257KB	88.745KB	0.24531	46.404	24.2001	31.2932

Поняття SNR, PSNR та MSE були розглянуті в п.4.3. Наглядні відмінності в самих зображеннях підтверджуються їхніми характеристиками.

6.5 Стиснення за допомогою мереж Хопфілда

Застосуємо модифікований метод BTC з використанням штучних мереж Хопфілда до тестової вибірки зображень. В подальшому будемо використовувати скорочення BTCNN (Block Truncation Coding Hopfield Network) для цього методу.

cameraman.tif



Рисунок 6.19 - Оригінальне
зображення



Рисунок 6.20 - Відновлене зображення

Lena.tif



Рисунок 6.21 - Оригінальне
зображення



Рисунок 6.22 - Відновлене зображення

fishingboat.tif



Рисунок 6.23 - Оригінальне
зображення



Рисунок 6.24 - Відновлене зображення

tank.tif



Рисунок 6.25 - Оригінальне
зображення



Рисунок 6.26 - Відновлене зображення

truck.tiff



Рисунок 6.27 - Оригінальне
зображення



Рисунок 6.28 - Відновлене зображення

Стиснення за допомогою запропонованого методу відбувається дещо довше, ніж за допомогою базового ВТС. Стиснення зображень за допомогою мереж Хопфілда для кожного зображення з тестової вибірки займає приблизно 6-8 секунд. Також час виконання залежить від розміру зображення. Так, для стиснення «cameraman.tif» знадобилось менше 5-и секунд. Зображення тестової вибірки мають розмір до 270 КВ, а отже стиснення зображень з високою роздільною здатністю розміром в кілька десятків мегабайт може зайняти досить значний час.

Втрата якості після стиснення для зображень cameraman.tif, fishingboat.tif, truck.tiff легкопомітна. Для двох інших зображень – tank.tiff, lena.tif – за суб'єктивним сприйняттям, якість зображення зберігається на значно кращому рівні, ніж після застосування базового ВТС. Кількісні характеристики наведені в таблиці 6.2.

Таблиця 6.2 - Кількісні характеристики зображень, стиснутих за допомогою ВТСНН

Зображення	Розмір до стиснення	Розмір після стиснення	Коеф. стиснення	MSE	SNR	PSNR
cameraman.tif	64KB	14.611KB	0.2283	78.8984	23.5777	29.0917
Lena.tif	278KB	78.21 6KB	0.22538	40.5131	24.9302	31.8131
fishingboat.tif	252KB	58.285KB	0.24129	49.7415	25.821	31.1636
tank.tif	256KB	56.819KB	0.26195	24.9268	29.2168	33.5685
truck.tif	257KB	56.494KB	0.23982	23.5632	27.1433	34.2365



7 АНАЛІЗ РЕЗУЛЬТАТІВ

Метод стиснення зображень ВТС, та ВТСН його модифікація, запропонована в даній роботі – методи стиснення з втратами. Тому на стиснутих зображеннях спостерігається розмитість та поява інших артефактів.

ВТС – це відомий метод для стиснення зображень з втратами, при якому зображення розбивається на блоки фіксованого розміру. Найбільш помітним артефактом при стисненні за допомогою ВТС є поява видимих контурів цих блоків. Зображення стиснуті за допомогою ВТСН менш схильні до цієї проблеми (хоча для невеликого за розміром зображення cameraman.tif їх чітко видно) і результуючі зображення є більш розмитими в порівнянні з оригіналом.

Алгоритм ВТСН також розбиває вхідне зображення на блоки, що помітно при збільшенні відновленого зображення.

Але вдалося зменшити вплив контурного ефекту і, за суб'єктивним сприйняттям, при масштабі 1:1 він є малопомітним.

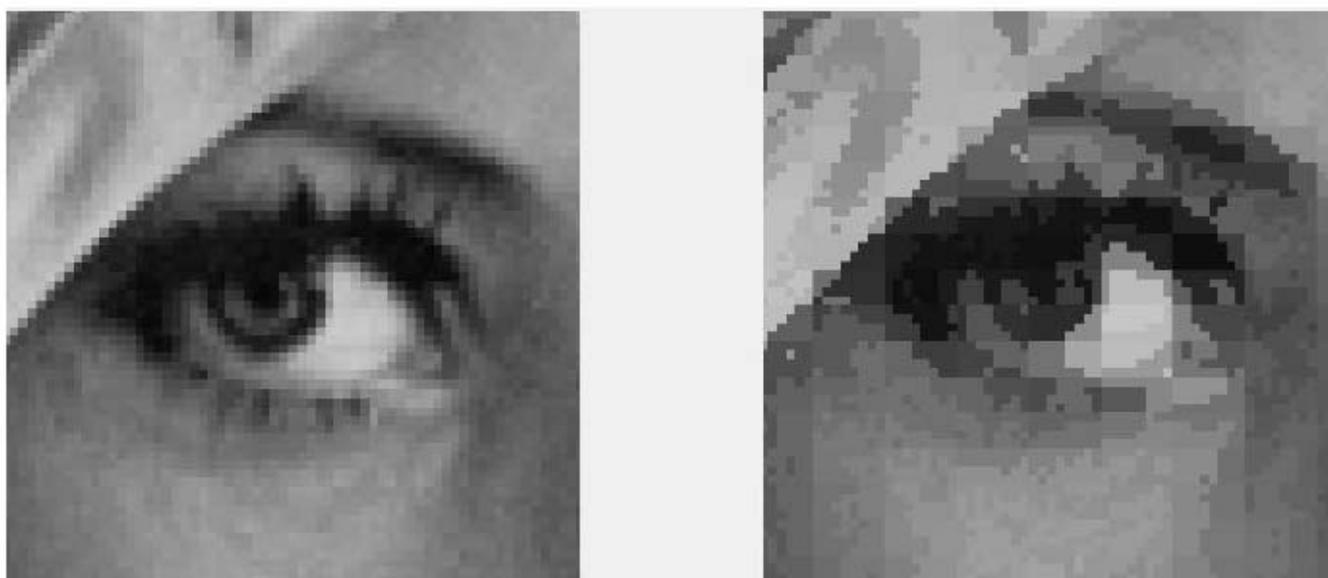


Рисунок 7.1 - Збільшені ділянки оригінального (зліва) та відновленого (справа) зображень

Розглянемо кількісні характеристики, наведені в табл. 6.1. та табл.6.2. Коефіцієнт стиснення сильно не відрізняються: для VTCHN варіюється в межах 0.220 – 0.262, тоді як для BTC - 0.228 - 0.270. Значного покращення в рівні стиснення досягнуто не було, так як в VTCHN змінено лише етап класифікації пікселів для кращої відповідності оригінальному зображенню, за допомогою мережі Хопфілда.

Для BTC властиве високе значення PSNR, при хороших показниках стиснення. При застосуванні VTCHN спостерігається покращення цієї характеристики на 9,41-9,615 %.

Результати стиснення для BTC за рівнем спадання по значенню похибки MSE виглядають наступним чином: cameraman.tif, fishingboat.tif, lena.tif, truck.tif, tank.tif. Для VTCHN зберігається такий же порядок, за виключенням truck.tif, значення MSE для якого менше, ніж у tank.tif. Загалом спостерігається значне покращення за цією характеристикою – її зменшення приблизно вдвічі. Так MSE для lena.tif стиснутого за допомогою BTC - 84.5126, за допомогою VTCHN - 40.5131. Це означає зменшення рівня неточностей у стиснутому зображенні, і те, що класифікація пікселів за допомогою мережі Хопфілда

класифікує пікселі ближче до значення оригіналу, ніж при класифікації за допомогою порівняння зі статистичними моментами.

В більшості випадків чим нижче значення MSE, тим вище значення SNR. Це твердження справедливе для всіх результатів, крім tank.tif, стиснутого за допомогою BTCN.

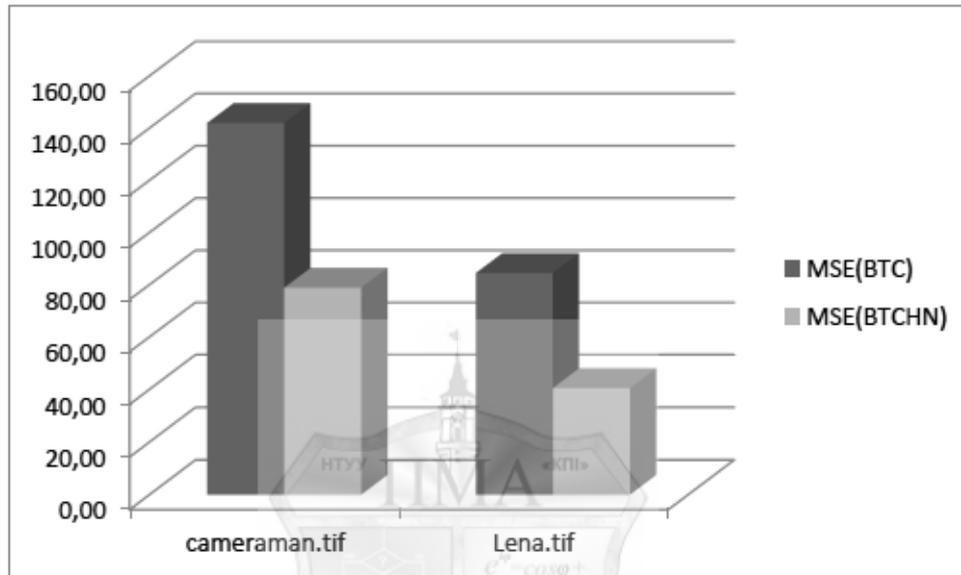


Рисунок 7.2 - Зміна значення MSE

Вище значення SNR відповідає кращій якості відновленого зображення. Для BTCHN також спостерігається покращення за цим параметром: найбільше змінилося значення для lena.tif збільшення в 1.146 рази, найменше для tank.tif – 1.098.

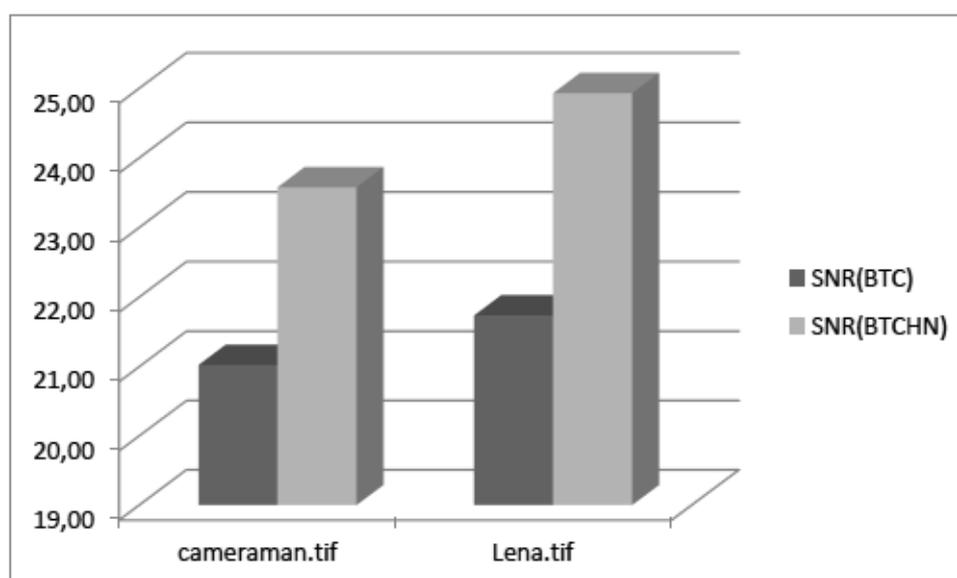


Рисунок 7.3 - Зміна значення SNR

Закономірним є також підвищення параметру PSNR, що видно з рисунку 7.4.

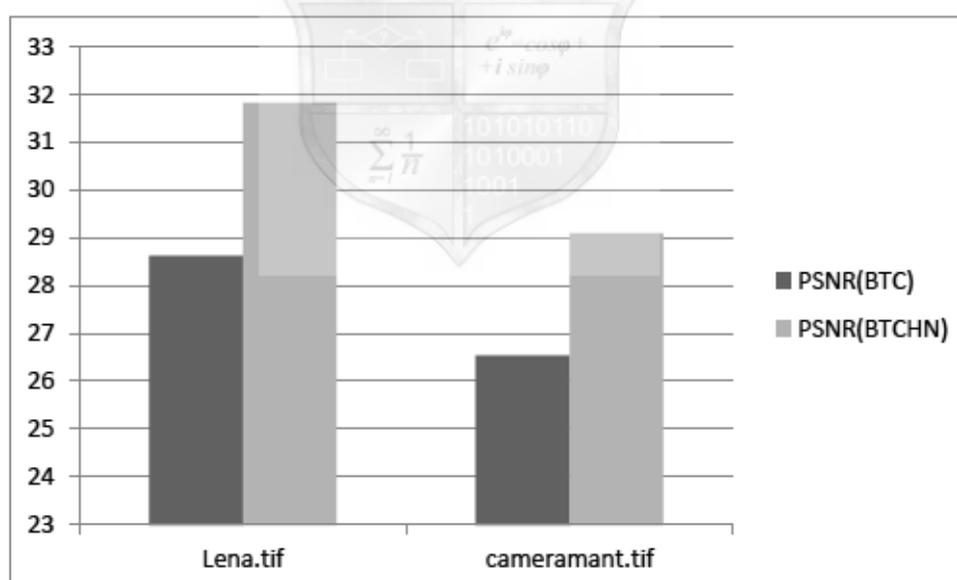


Рисунок 7.4 - Зміна значення PSNR

Отже, за рахунок удосконалення методу класифікації пікселів в межах блоку було досягнуто кращу якість зображення при збереженні рівня стиснення. Це помітно при сприйняттєвому порівнянні зображень відновлених

алгоритмами BTC та BTCHN, а також підтверджується кількісними показниками MSE, SNR і PSNR.



ВИСНОВКИ

Дана магістерська дисертація присвячена проблемі стиснення цифрових зображень та розробка нового методу в цій сфері, в якому використовуватимуться штучні нейронні мережі. На меті було провести дослідження штучних нейронних мереж як інструменту для стиснення даних та запропонувати новий метод стиснення зображень, який би базувався на цьому підході.

Перш за все, було вивчено основні принципи стиснення зображень, та детально розглянуто деякі відомі алгоритми. Одним з них є алгоритм Block Truncation Coding.

Було розглянуто існуючі алгоритми нейромережевого стиснення даних, і як виявилось в ході проведення обчислювального експерименту - вони є малоефективними самі по собі, та не завжди придатними для вхідних даних. Тому було прийнято рішення використовувати штучні нейронні мережі як один з етапів алгоритму стиснення чи використовувати у комбінації з іншими методиками.

Запропонований в даній дисертації алгоритм Block Truncation Coding - Hopfield Network - базується на алгоритмі BTC та використовує штучні мережі Хопфілда. Даний тип штучних нейронних мереж був розроблений для реалізації асоціативної пам'яті та відновлення пошкоджених образів. Але в цій роботі властивості мережі використовуються для іншої задачі - квантування пікселів зображення.

Більшість кроків алгоритму BTCNN такі ж, як і в BTC. Відмінність полягає в способі пошуку порогового значення. В BTC рахується статистичний момент блоку і використовується в якості порогового значення, і пікселі блоку,

значення яких більше рівне порогового встановлюються в «1», інакше в «0». Але в запропонованому методі VTCHN значення пікселів визначається стійким станом мережі Хопфілда. Стійкість стану визначається виходячи з поняття енергії штучної нейронної мережі – в стійкому стані вона досягає свого мінімального значення, а значення нейронів мережі перестають змінюватися.

Стиснення за допомогою реалізованого методу VTCHN відбувається довше, за рахунок виконання великої кількості ітерації до досягнення стійкого стану. Стиснення зображень розміром 250-270KB відбувається триває 6-8 секунд, і це значення зростає з розміром вхідного зображення.

Рівень стиснення для базового і модифікованого BTC значно не відрізняється: для розміру блоку 4×4 коефіцієнт стиснення варіюється в межах 0.22-0.25 для обох методів.

Результати отримані за допомогою методу VTCHN демонструють кращу якість зображень. Хоча, в деяких випадках проявляються, як і в BTC, контурні артефакти. Тому що в кожному блоці наявні лише два рівні відновлення, і під час стиснення значна частина інформації відкидається. Але частота появи таких артефактів є нижчою і за суб'єктивним сприйняттям - менш помітною.

VTCHN дає вигравш порівняно з BTC по всім розглянутим кількісним параметрам якості відновленого зображення - MSE, SNR, PSNR.

Алгоритм було розроблено і протестовано для монохромних зображень. В подальшому можна розширити алгоритм для роботи з кольоровими зображеннями. Для подальшої можливості застосування запропонованого алгоритму прикладними програмами на практиці слід розглянути доповнення алгоритму VTCHN існуючими методами стиснення з втратами та без, та представити у вигляді формату збереження зображень.

Також можна провести випробування ефективності застосування VTCHN замість BTC у вже існуючих алгоритмах, таких як JPEG.

Результати досліджень нейромережевого підходу до стиснення даних в ході виконання даної дисертації було оформлено і представлено на

міжнародній науковій конференції ім. Таран Т.А. “Інтелектуальний аналіз інформації – ІАІ 2015” та VII конференції магістрантів та аспірантів “Прикладна математика та комп’ютинг - ПМК 2015”.



СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. *Сэломон, Дж.* Сжатие данных, изображений и звука [Текст] / Дж. Сэломон : пер. с англ. – Москва : Техносфера, 2004. – 366 с.
2. Rabbani M. Digital Image Compression Techniques [Text] / M. Rabbani, P.W. Jones. - Bellingham, Washington : SPIE Press, 1991.
3. Foley J. Computer Graphics - Practice and Principles [Text] / J. Foley, M. van Dam, R. Feiner. - Addison Wesley, 1993.
4. *Murray J.D.* Encyclopedia of Graphics File Formats, 2nd Edition [Text] / J.D. Murray, W. van Ryper. – Sebastopol, California : O'Reilly Media, 1996. – 1152 p.
5. *Werbos P.* Backpropagation through time. What it does and how to do it [Text] / Proc. IEEE, vol.78, NO. 10. – 1990. – pp. 1550-1560.
6. *Бодянский Е.В.* Искусственные нейронные сети: архитектуры, обучение, применения [Текст] / Е.В. Бодянский, О.Г. Руденко – Харьков : Телетех, 2004. – 369 с.
7. Verma B. Aneural Network based Technique for Data Compression [Text] / B. Verma, M. Blumenstein, S. Kulkarni. - Proceedings of the IASTED International Conference on Modelling and Simulation, MSO97, Singapore, 1997.
8. *Хайкин С.* Нейронные сети. Полный курс [Текст] / С. Хайкин. – Москва : Вильямс, 2006. – 1104 с.
9. *Олефір А.О.* Нейромережева модель стиснення даних [Текст] / А.О. Олефір, М.В. Превір. – Международная конференция имени Т.А. Таран «Интеллектуальный анализ информации». – Київ : Просвіта, 2015 – с.152 – 157.

10. *Олефір А.О.* Застосування штучних нейронних мереж типу перцептрон та мереж Хопфілда для стиснення даних [Текст] / А.О. Олефір, М.В. Превір. – Прикладна математика та комп'ютинг 2015. Збірник тез доповідей. – Кнів : Просвіта, 2015. – с. 211-216.
11. Rojas R. Neural Networks [Text] / R. Rojas Berlin : Springer-Verlag, 1996.
12. *Delp E.J.* Image Coding Using Block Truncation Coding [Text] / E.J. Delp, O.R. Mitchell. – IEEE Transactions on Communications, 27, pp. 1335-1342.
13. *Udpikar V.R.* BTC image coding using vector quantization [Text] / V.R. Udpikar, J.P. Raina. - Trans. Commun., March 1987, vol. COM-35, pp. 352-356.
14. *Udpikar V.R.* Modified algorithm for block truncation coding of monochrome images [Text] / V.R. Udpikar, J.P. Raina. - Electronic Letters, 1985, Vol.21, No.20, pp. 900-902.
15. *Hopfield J.J.* Neural networks and physical systems with emergent collective computational abilities [Text] / J.J. Hopfield. - Proceedings of National Academy of Sciences, vol. 79 no. 8, April 1982, pp. 2554—2558.
16. *Qiu G.* Variable bitrate block truncation coding for image compression [Text] / G.Qiu, M.R. Varley, T.J. Terrell - Proc. 3rd International Conference on Artificial Neural Networks, Brighton, May 1993, pp233-237.
17. Casella G. Theory of Point Estimation [Text] / G. Casella, E.L. Lehman. - New York : Springer-Verlag, 1999.
18. Franti P. Compression of Digital Images by Block Truncation Coding [Text] / P. Franti, O. Nevalainen, T. Kaukoranta. - The Computer Journal, 37(4), 1994, pp. 308-332.
19. *Computer Graphics Data* [Електронний ресурс]. – Режим доступу: <http://graphics.cs.williams.edu/data/images.xml>