

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ДЛЯ ПЕРЕТВОРЕННЯ ТА СТИСНЕННЯ ЗОБРАЖЕНЬ
IMAGE CONVERSION AND
COMPRESSION SOFTWARE

Виконав(ла): студент(ка) 4 курсу, групи КНТ-113сп
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ПОПОВ Д.А.

(ПРИЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОРОТКИЙ О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПОПОВА Данила Андрійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для перетворення та стиснення зображень. Image conversion and compression software

керівник проєкту (роботи) к.т.н., доцент, ШИТІКОВА Олена Вікторівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ШИТІКОВА О.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Данило ПОПОВ
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Олена ШИТІКОВА
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
89 с., 3 табл., 20 рис., 3 дод., 20 джерел.

C#, VISUAL STUDIO, ЗОБРАЖЕННЯ, МОВА ПРОГРАМУВАННЯ,
ОБРОБКА ЗОБРАЖЕНЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для обробки зображень.

Предмет дослідження – програмні засоби для перетворення зображень.

Мета роботи – розробка програмного забезпечення для перетворення та стиснення зображень, що забезпечує скорочення витрат дискового простору та мережевих ресурсів при обробці графічних даних.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Visual Studio.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для перетворення та стиснення зображень. Створено програмне забезпечення для перетворення та стиснення зображень. Проведено тестування розробленої програмної системи для перетворення та стиснення зображень.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для перетворення та стиснення зображень за допомогою мови програмування C# та середовища розробки Visual Studio.

Галузь використання – обробка зображень.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 3 tables, 20 figures, 3 appendixes, 20 sources.

C#, VISUAL STUDIO, IMAGE, PROGRAMMING LANGUAGE, IMAGE PROCESSING, SOFTWARE.

The object of research is the process of developing software for image processing.

The subject of the research is software for image conversion.

The purpose of this work is to develop the software for image conversion and compression, which provides a reduction in disk space and network resources when processing graphic data.

Materials, methods and technical tools: C# programming language, Visual Studio development environment.

Results. Project solutions for the creation of software for image conversion and compression has been designed. The software to support the process for image conversion and compression has been developed. The developed software system for image conversion and compression has been tested.

Conclusions. The goal of the work has been achieved. The software tools for image conversion and compression using the C# programming language and the Visual Studio development environment have been developed.

Scope of use – image processing.

ЗМІСТ

	С.
Вступ.....	8
1 Аналіз предметної області	11
1.1 Програмне забезпечення для перетворення та стиснення зображень.....	11
1.2 Аналіз програмного забезпечення для перетворення та стиснення зображень	18
1.3 Висновки за розділом 1	29
2 Матеріали і методи	32
2.1 Вибір мови програмування.....	32
2.2 Вибір середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень.....	35
2.3 Висновки за розділом 2	39
3 Опис програми	41
3.1 Структура програмного забезпечення для перетворення та стиснення зображень	41
3.2 Функціонування програмного забезпечення для перетворення та стиснення зображень	43
3.3 Реалізація алгоритмів обробки зображень	45
3.4 Проєктування інтерфейсу програмного забезпечення для перетворення та стиснення зображень	49
3.5 Висновки за розділом 3	52
4 Експлуатація, тестування та експериментальне дослідження програми.....	54
4.1 Призначення й умови застосування програми	54
4.2 Характеристики програми для перетворення та стиснення зображень ..	56
4.3 Інструкція по експлуатації програми.....	57
4.3.1 Звернення до програми.....	57
4.3.2 Вхідні й вихідні дані.....	57
4.3.3 Повідомлення.....	58

4.4 Виконання програмного забезпечення для перетворення та стиснення зображень	58
4.5 Тестування програмного забезпечення для перетворення та стиснення зображень	60
4.6 Висновки за розділом 4	60
Висновки.....	61
Перелік джерел посилання	65
Додаток А Технічне завдання.....	67
Додаток Б Фрагмент тексту програми	71
Додаток В Слайди презентації	83

ВСТУП

Актуальність програмного забезпечення для перетворення та стиснення зображень зумовлюється стрімким зростанням обсягів візуальної інформації, що генерується та передається в сучасному цифровому середовищі. У процесі розвитку інформаційних технологій спостерігається постійне підвищення роздільної здатності графічних даних, що, у свою чергу, призводить до збільшення вимог до обчислювальних ресурсів, пропускної здатності мереж та обсягів сховищ. За таких умов виникає потреба у застосуванні ефективних методів обробки зображень, які дозволяють зменшувати їхній розмір без суттєвої втрати якості або адаптувати формат представлення відповідно до конкретних задач використання [1]-[3].

Особливої значущості набувають програмні засоби перетворення зображень у контексті забезпечення сумісності між різними платформами, пристроями та програмними середовищами. Різноманітність форматів графічних файлів і специфічні вимоги до їх відображення чи подальшої обробки обумовлюють необхідність автоматизованих інструментів, здатних виконувати конвертацію із збереженням ключових характеристик зображення. У цьому аспекті програмні рішення виконують роль універсального посередника між різними стандартами представлення даних, що забезпечує гнучкість та масштабованість інформаційних систем [2]-[4].

Значна роль програмного забезпечення такого типу простежується і в галузях, пов'язаних з обробкою великих масивів даних, зокрема в машинному навчанні, комп'ютерному зорі та аналітиці. Передобробка зображень, яка включає зміну розміру, нормалізацію, фільтрацію та стиснення, є невід'ємним етапом підготовки даних до подальшого аналізу. Оптимізація цих процесів дозволяє підвищити ефективність алгоритмів та зменшити витрати часу на обчислення [4]-[7].

Отже, необхідність створення та вдосконалення програмних засобів для перетворення і стиснення зображень визначається комплексом факторів, серед

яких зростання обсягів даних, потреба у швидкій передачі інформації, забезпечення сумісності форматів та підвищення ефективності обробки. У сучасних умовах такі системи виступають важливим елементом інфраструктури цифрових технологій, сприяючи оптимізації використання ресурсів і покращенню якості інформаційних сервісів [3]-[7].

Недоліки існуючого програмного забезпечення для перетворення та стиснення зображень проявляються у низці аспектів, пов'язаних як з технічними обмеженнями алгоритмів, так і з особливостями їх практичної реалізації. У процесі обробки графічних даних часто спостерігається компроміс між ступенем стиснення та якістю зображення, внаслідок чого при інтенсивному зменшенні обсягу файлу виникають втрати деталей, поява артефактів і спотворення кольорів. Це особливо критично у випадках, коли зображення використовуються для професійних або наукових цілей, де точність відображення має принципове значення. Суттєвим недоліком є також залежність ефективності стиснення від типу зображення та обраного формату. Алгоритми, що добре працюють для фотографій, можуть виявитися неефективними для графіки з чіткими контурами або текстом, і навпаки. У результаті виникає необхідність додаткового налаштування параметрів або використання різних інструментів для різних задач, що ускладнює процес автоматизації та знижує універсальність програмного забезпечення. Не менш важливим є питання сумісності та стандартизації. Існування великої кількості форматів і варіацій їх реалізації призводить до ситуацій, коли певні програми некоректно обробляють окремі файли або не підтримують новітні стандарти. Це створює додаткові труднощі при інтеграції різних програмних систем та обміні даними між ними [1]-[7].

Тому актуальною є розробка програмного забезпечення для перетворення та стиснення зображень. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для перетворення та стиснення зображень, що забезпечує

скорочення витрат дискового простору та мережевих ресурсів при обробці графічних даних.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для перетворення та стиснення зображень;
- здійснити проектування програмного забезпечення для перетворення та стиснення зображень;
- створити програмне забезпечення для перетворення та стиснення зображень;
- виконати тестування розробленого програмного забезпечення для перетворення та стиснення зображень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для перетворення та стиснення зображень

Програмне забезпечення для перетворення та стиснення зображень являє собою клас програмних систем, призначених для обробки графічних файлів з метою зміни їх формату, розміру або зменшення обсягу зберігання без значної втрати якості. Такі програмні системи дозволяють оптимізувати зображення для подальшого використання у вебсайтах, мультимедійних проєктах або базах даних, підвищуючи швидкість завантаження та економлячи ресурси сховища. В їх основі лежать алгоритми обчислювальної обробки, що аналізують піксельні дані, визначають повторювані або непотрібні елементи та ефективно перетворюють інформацію у стислий формат [1], [2].

До характерних особливостей таких програм належить можливість роботи з різними форматами зображень, включаючи популярні растрові та векторні типи, а також підтримка пакетної обробки великої кількості файлів одночасно. Програмне забезпечення може надавати інструменти для вибору ступеня стиснення, що дозволяє контролювати баланс між розміром файлу та візуальною якістю, а також функції для перетворення формату зображень, наприклад, з JPEG у PNG або WebP, що забезпечує сумісність із різними платформами. Часто такі системи інтегруються з вебсайтами або системами управління вмістом, надаючи можливість автоматичної обробки медіафайлів без додаткових дій користувача [1], [2].

Переваги подібного програмного забезпечення полягають у значному скороченні обсягу файлів без втрати візуальної інформації, підвищенні продуктивності вебресурсів та зменшенні витрат на зберігання даних. Системи забезпечують зручний інтерфейс, що спрощує роботу з великим числом зображень, дозволяючи швидко завантажувати, обробляти та отримувати готові результати. Крім того, розширені можливості для інтеграції та автоматизації робочих процесів роблять такі програмні продукти гнучким

інструментом як для окремих користувачів, так і для корпоративного використання. Завдяки цим особливостям програмне забезпечення для перетворення та стиснення зображень стає невід'ємною частиною сучасних цифрових робочих процесів, забезпечуючи ефективне управління графічними ресурсами та оптимізацію їх використання [1], [2].

Стиснення зображень являє собою технологічний процес, спрямований на зменшення обсягу файлів графічних матеріалів. Воно реалізується через оптимізацію даних зображення, що може включати видалення частини інформації або застосування спеціальних алгоритмів, які перепакують файл у більш компактному форматі без значної втрати якості. Подібна обробка дозволяє економити простір для зберігання та прискорює відтворення графіки у різних цифрових середовищах, забезпечуючи більш плавну взаємодію користувача з вебресурсами чи програмними продуктами. Стиснення стає ключовим елементом процесу оптимізації зображень та підвищення ефективності роботи вебсайтів і застосунків [3], [4].

Важливість стиснення полягає у тому, що зменшені файли відкриваються набагато швидше, що прямо впливає на продуктивність ресурсів, показники взаємодії користувачів та пошукову оптимізацію. Прискорене завантаження зображень покращує загальний досвід користувача, знижує час очікування та сприяє більш ефективному відображенню контенту. Крім того, швидкість завантаження відіграє значну роль у комерційних аспектах вебресурсів, таких як коефіцієнт конверсії та утримання аудиторії, що робить стиснення зображень стратегічно важливою процедурою для розробників [3], [4].

Зазвичай стиснення поєднується з іншими технологіями оптимізації продуктивності, що дозволяють підвищити швидкість доставки контенту. Використання систем кешування, дозволяє ефективно розподіляти дані між серверами та кінцевими користувачами, а балансування навантаження знижує ризик перевантаження вебсерверів. Техніки відкладеного завантаження зображень дають змогу першочергово демонструвати основний контент,

забезпечуючи більш швидке відображення сторінок. Проте серед усіх методів поліпшення продуктивності стиснення зображень залишається одним із найшвидших і найбільш ефективних способів впливу на загальну швидкодію вебресурсів [3], [4].

Стиснення зображень з втратами являє собою процес зменшення обсягу файлу графічного матеріалу шляхом відбору найсуттєвішої інформації та усунення менш помітних деталей. Це можна порівняти з подорожнім, який обирає лише необхідні речі для поїздки, відмовляючись від усіх зайвих предметів гардероба. Подібним чином, під час перегляду зображень у більшості цифрових середовищ не потрібна максимальна роздільна здатність чи всі пікселі файлу; досить тих даних, які забезпечують сприйняття картинки без втрати візуальної цілісності для користувача. Такий підхід дозволяє значно зменшити обсяг файлу, роблячи його більш економним для зберігання та передачі через мережі без помітного погіршення якості відображення [3], [4].

Принцип роботи стиснення з втратами полягає у виборі основних елементів зображення, які несуть найбільше сенсове навантаження, та відкиданні менш важливої інформації. Використовуються спеціальні алгоритми, які перезаписують дані файлу, зменшуючи його розмір, але залишаючи візуально прийнятне відтворення. В результаті отримується файл, що складається з меншої кількості байтів і швидше передається через мережі або завантажується на пристрої користувачів [3], [4].

Серед найбільш поширених методів стиснення з втратами використовуються формати, які оптимізовані для вебконтенту. Формат JPEG, створений для ефективного зменшення обсягу фотографій, дозволяє отримати значну економію розміру файлу при мінімальній видимій втраті якості. Інший підхід пропонує подібну функціональність із додатковою підтримкою нових технологій стиснення та ефективнішою оптимізацією для інтернет-контенту. Високоєфективний формат HEIF, забезпечує сучасне стиснення, дозволяючи зберігати більше деталей у компактнішому файлі. Усі ці методи працюють на основі зменшення кількості даних, що зберігаються в кожному пікселі, при

цьому зберігаючи основну структуру та сприйняття зображення користувачем [3], [4].

Стиснення зображень без втрат являє собою процес оптимізації файлу зображення, при якому жодна інформація не втрачається, а розмір файлу зменшується за рахунок переформатування та перезапису даних за допомогою математичних алгоритмів. У цьому випадку цифрове зображення після обробки виглядає практично ідентично оригіналу, але займає менше місця на диску. Механізм можна порівняти з упаковкою речей, коли об'єкти акуратно розташовуються у валізі так, щоб не змінювати їх сутність, але водночас економити простір. Цей метод є корисним для збереження високої якості графічного матеріалу, коли важливі всі деталі зображення, і неможливо допустити видалення будь-яких пікселів [3], [4].

Хоч стиснення без втрат дозволяє скоротити обсяг файлів, воно зазвичай не дає такого значного зменшення розміру, як методи зі втратами, і тому його ефективність у вебоптимізації може бути нижчою. Розробники сайтів та програмного забезпечення повинні оцінювати баланс між якістю та швидкістю завантаження для кінцевого користувача, підбираючи відповідний метод стиснення залежно від конкретного випадку використання та потреб аудиторії [3], [4].

Для реалізації стиснення без втрат застосовуються різні алгоритми та формати файлів, що дозволяють перетворювати та кодувати пікселі ефективним чином. Формати PNG і GIF забезпечують збереження всіх вихідних даних зображення і широко використовуються в інтернет-середовищі для графіки та анімацій. Інші формати, такі як BMP чи RAW, через великий обсяг файлів потребують конвертації перед застосуванням у веб-середовищі, оскільки їх пряме використання нераціональне. Алгоритми кодування, включно з методами кодування по довжині, арифметичним кодуванням, Хаффманом, LZW та Flate/deflate, дозволяють ефективно організувати дані всередині файлу, зменшуючи обсяг без втрати будь-яких значущих пікселів. Деякі з цих алгоритмів орієнтовані на повторювані

елементи, інші – на частотний розподіл символів, що дозволяє оптимізувати структуру файлу й забезпечити швидку обробку та зберігання зображень без шкоди для їхньої візуальної якості [3], [4].

Процес оптимізації розміру цифрових зображень включає кілька взаємопов'язаних етапів. Спочатку необроблене зображення обробляється для створення більш компактного файлу, що зменшує його обсяг на диску. Ця операція називається кодуванням, а програмне забезпечення, яке виконує перетворення вихідного зображення у стиснутий формат, вважається кодером. Наступний етап передбачає перетворення стислого файлу назад у зображення, яке відтворює візуальні характеристики оригіналу, відоме як реконструкція. Ця дія здійснюється за допомогою декодування, а програмне забезпечення, що відновлює зображення, називається декодером. Поєднання кодувального та декодувального модуля формує кодек, який забезпечує повний цикл обробки цифрових зображень [4], [5].

Стиснення цифрових зображень завжди передбачає баланс між швидкістю обробки даних і візуальною якістю кінцевого файлу. Головне завдання алгоритмів оптимізації полягає в тому, щоб розміщувати бітову інформацію таким чином, аби мінімізувати сприймані спотворення, залишаючи ключові деталі зображення незмінними. Для оцінки результатів використовуються різні показники, які вимірюють співвідношення між розміром файлу та збереженою якістю, включно з базовими метриками на кшталт пікового співвідношення сигнал/шум, структурної подібності, а також більш складними перцептивними показниками, що моделюють сприйняття людським оком. Алгоритми враховують особливості зорового сприйняття: менш помітні помилки в текстурованих або високочастотних ділянках зображення можуть бути допущені, що дозволяє досягати кращого співвідношення обсягу файлу та якості [4], [5].

Різноманіття методів стиснення породжує потребу в стандартизації. Використання власних кодеків для обміну файлами потребує забезпечення доступності відповідного декодера для всіх користувачів, які отримують

стиснені зображення. Встановлення стандартів забезпечує сумісність між різними платформами та дозволяє використовувати спеціалізоване обладнання для підвищення продуктивності обробки. Найпоширенішим прикладом такої стандартизації є формат JPEG, який з моменту свого створення став основним для природних растрових зображень та цифрової фотографії [4], [5].

Алгоритми перетворення та стиснення зображень створені для зменшення обсягу даних, що зберігають візуальну інформацію, без значної втрати сприйманої якості. У процесі обробки цифрове зображення спочатку перетворюється в інший просторово-частотний формат, який дозволяє більш ефективно кодувати його пікселі. Таке перетворення забезпечує виділення суттєвої інформації про структуру та деталі зображення, одночасно зменшуючи надлишковість даних. Відмінною рисою цих алгоритмів є здатність концентрувати енергію сигналу у невеликій кількості коефіцієнтів, що дозволяє проводити подальше стискання без значного погіршення візуальної якості [4], [5].

Одним із методів перетворення, що широко використовується у сучасних алгоритмах стиснення, є вейвлетне перетворення. Вейвлети дозволяють аналізувати зображення на різних рівнях масштабів, розкладаючи його на високочастотні та низькочастотні компоненти. Це дає можливість ефективно виділяти деталі та текстури, які є важливими для сприйняття, і одночасно піддавати менш важливі частини більш сильному стисненню. Вейвлетне представлення дозволяє здійснювати багаторівневу декомпозицію зображення, що робить алгоритм гнучким і придатним для компресії різних типів зображень, від фотографій до графічних схем [6], [7].

Серед найвідоміших вейвлетів для стиснення зображень виділяються вейвлети Хаара та Добеши. Вейвлет Хаара відзначається простотою реалізації та високою швидкістю обробки. Його алгоритм використовує пряме та зворотнє перетворення, розбиваючи зображення на блоки та обчислюючи різниці та середні значення пікселів. Це дозволяє виділити ключові характеристики зображення та скоротити обсяг даних, при цьому зберігаючи

загальну форму та контури об'єктів. Вейвлет Хаара особливо ефективний для зображень із чіткими контрастними межами, таких як графіки та текстури з різкими переходами, але може поступатися у передачі плавних градієнтів в зображеннях [6], [7].

Вейвлети Добеши, на відміну від Хаара, забезпечують більш складну структуру функцій з гладкішими переходами. Вони дозволяють краще зберігати тонкі градієнти та деталізацію природних сцен, що робить їх оптимальним вибором для фотографій і реалістичних зображень. Алгоритм Добеши розкладає сигнал на декілька рівнів з використанням спеціально підібраних базисних функцій, які володіють високою локалізацією у часі та частоті. Це дозволяє зберігати великі об'єми корисної інформації при одночасному видаленні надлишкових даних, що ефективно зменшує розмір файлу [6], [7].

Застосування цих вейвлетів у стисненні зображень часто поєднується з подальшою квантизацією та кодуванням залишкових коефіцієнтів, що підвищує ефективність алгоритму. Вейвлетне перетворення дає змогу виконувати багаторівневе стиснення, автоматично виділяючи найважливіші візуальні елементи для збереження та менш значущі деталі для агресивного скорочення. Завдяки такому підходу досягається висока компресія без помітного погіршення якості для кінцевого користувача, що робить вейвлети Хаара та Добеши невід'ємною частиною сучасних методів оптимізації зображень [6], [7].

Визначено, що програмне забезпечення для перетворення та стиснення зображень являє собою клас програмних систем, призначених для обробки графічних файлів з метою зміни їх формату, розміру або зменшення обсягу зберігання без значної втрати якості. Такі програмні системи дозволяють оптимізувати зображення для подальшого використання у вебсайтах, мультимедійних проєктах або базах даних, підвищуючи швидкість завантаження та економлячи ресурси сховища.

1.2 Аналіз програмного забезпечення для перетворення та стиснення зображень

Проаналізуємо програмне забезпечення для перетворення та стиснення зображень [8]-[16].

Програмне забезпечення EWWW Image Optimization (рис. 1.1) є ефективним інструментом для обробки та оптимізації зображень, який орієнтований на використання в середовищі веброзробки. Його особливістю є виконання всіх операцій безпосередньо на стороні сервера користувача, що дозволяє уникнути залежності від сторонніх сервісів та не потребує додаткової реєстрації чи отримання спеціальних ключів доступу. Такий підхід забезпечує більший контроль над процесом обробки даних і підвищує рівень конфіденційності [9], [10], [14].

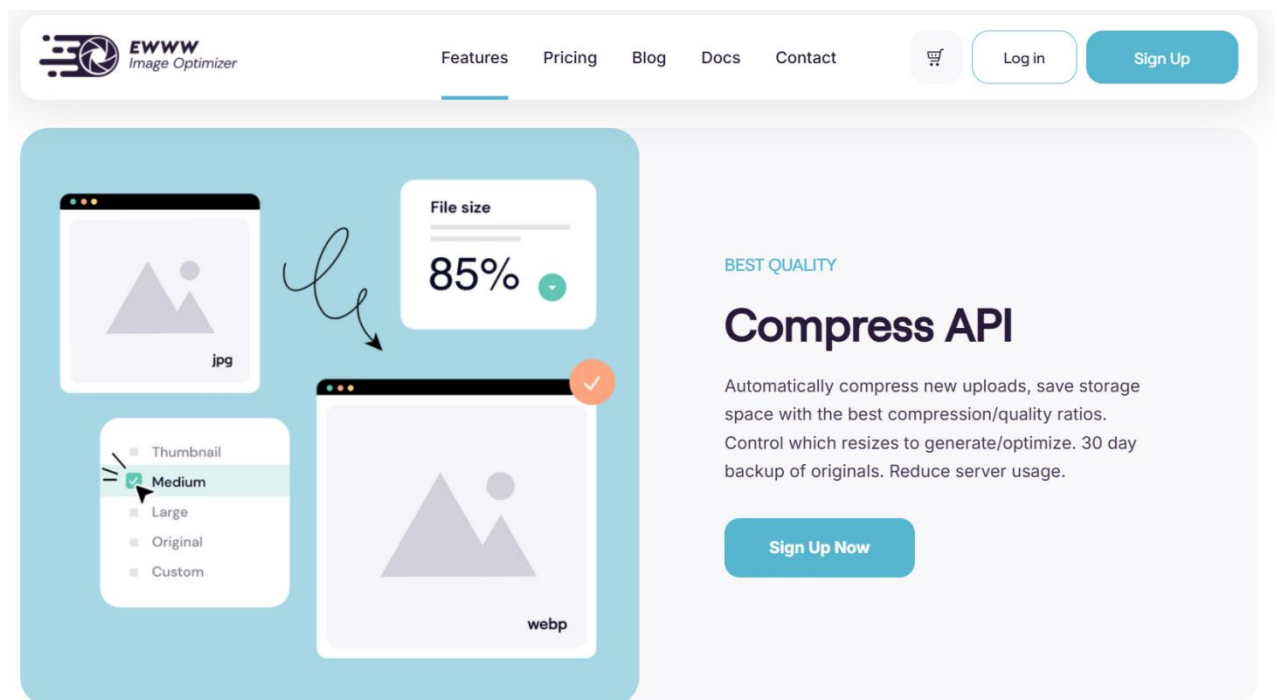


Рисунок 1.1 – Програмне забезпечення EWWW Image Optimization [14]

Функціональні можливості цього програмного забезпечення охоплюють широкий спектр операцій, пов'язаних зі стисненням зображень. Реалізовано як режими зі збереженням початкової якості, так і варіанти, що

передбачають часткову втрату даних задля досягнення більшого рівня зменшення обсягу файлів. У межах роботи застосовується адаптивний підхід до вибору формату, що дозволяє автоматично підбирати найбільш доцільний варіант представлення графічних даних залежно від їхніх характеристик. Це сприяє досягненню оптимального балансу між якістю зображення та його розміром [9], [10], [14].

Особливу увагу приділено автоматизації процесу обробки. Після завантаження нових файлів у систему їх оптимізація може виконуватися без додаткового втручання користувача, що значно спрощує робочий процес. Крім того, передбачено можливість масової обробки вже наявних зображень, що є актуальним у випадках роботи з великими обсягами графічних даних. Обробка може здійснюватися не лише для файлів, що зберігаються у стандартних каталогах, але й для інших ресурсів, що розширює сферу застосування програмного засобу [9], [10], [14].

Важливою характеристикою EWWW Image Optimization є підвищення швидкості завантаження вебсторінок за рахунок оптимізації структури зображень. Це досягається завдяки застосуванню прогресивного відображення, що дозволяє користувачу бачити попередній варіант зображення ще до повного завершення його завантаження. Таким чином покращується загальне сприйняття швидкодії вебресурсу [9], [10], [14].

Поряд із базовими можливостями, у EWWW Image Optimization передбачено розширений функціонал, доступ до якого надається у межах додаткових сервісів. У цьому випадку досягається більш глибокий рівень стиснення, а також відкриваються додаткові інструменти для користувачів із підвищеними вимогами до обробки зображень. Це робить програмне забезпечення гнучким рішенням, придатним як для базових, так і для більш складних сценаріїв використання [9], [10], [14].

Разом із перевагами слід враховувати і певні особливості функціонування. Наприклад, окремі службові дані, що створюються під час роботи, можуть не видалятися автоматично після припинення використання

програмного засобу, що потребує додаткової уваги з боку користувача. Крім того, базові можливості EWWW Image Optimization обмежуються менш агресивними методами стиснення, що може впливати на кінцевий результат у порівнянні з розширеними варіантами обробки [9], [10], [14].

Загалом програмне забезпечення EWWW Image Optimization являє собою потужний інструмент для оптимізації зображень, який поєднує гнучкість налаштувань, автономність виконання та високу ефективність. Його використання доцільне у випадках, коли необхідно забезпечити контроль над процесом обробки графічних даних і досягти покращення продуктивності вебресурсів без залучення зовнішніх сервісів [9], [10], [14].

Основними характеристиками програмного забезпечення EWWW Image Optimization є такі [9], [10], [14]:

- локальна обробка зображень: оптимізація виконується безпосередньо на сервері користувача без необхідності передачі даних до зовнішніх сервісів, що забезпечує підвищений рівень контролю та конфіденційності [9], [10], [14];
- підтримка різних методів стиснення: у EWWW Image Optimization реалізовано як варіанти без втрати якості, так і режими зі зменшенням точності для досягнення більшого рівня компресії [9], [10], [14];
- автоматизація обробки: у програмі EWWW Image Optimization нові зображення можуть оптимізуватися автоматично під час завантаження без додаткових дій з боку користувача [9], [10], [14];
- масова оптимізація файлів: у EWWW Image Optimization передбачено механізм одночасної обробки великої кількості вже існуючих зображень, що спрощує роботу з великими наборами даних [9], [10], [14];
- адаптивне перетворення форматів: забезпечується автоматичний вибір найбільш доцільного формату зображення залежно від його характеристик та умов використання [9], [10], [14];
- підтримка сучасних форматів: у EWWW Image Optimization надається можливість перетворення зображень у більш ефективні формати, що сприяє зменшенню розміру файлів без суттєвої втрати якості [9], [10], [14];

- оптимізація поза стандартною бібліотекою: обробка може виконуватися не лише для файлів у медіатеці, але й для інших зображень, що використовуються у системі [9], [10], [14];
- підвищення швидкодії вебресурсів: використання оптимізованих зображень у програмі EWWW Image Optimization сприяє швидшому завантаженню сторінок та покращенню користувацького досвіду [9], [10], [14];
- гнучкість налаштувань: у програмному забезпеченні EWWW Image Optimization передбачено широкий набір параметрів, що дозволяють адаптувати процес оптимізації відповідно до конкретних вимог [9], [10], [14];
- можливість розширення функціоналу: додаткові сервіси програмного забезпечення EWWW Image Optimization відкривають доступ до більш глибокого стиснення та розширених можливостей обробки [9], [10], [14].

Серед недоліків програмного забезпечення EWWW Image Optimization можна відзначити залежність ефективності роботи від ресурсів сервера, що може впливати на швидкість обробки при великих обсягах даних, а також обмежені можливості базової версії у порівнянні з розширеними варіантами, які потребують додаткових витрат. Крім того, окремі службові дані можуть залишатися у системі після припинення використання програмного засобу, що потребує ручного очищення, а налаштування великої кількості параметрів інколи ускладнює процес початкового освоєння [9], [10], [14].

Програмне забезпечення ImageRecycle (рис. 1.2) представляє собою інструмент для оптимізації графічних ресурсів, який орієнтований на спрощення процесу зменшення розміру файлів без суттєвого впливу на їх візуальну якість. У межах його використання забезпечується обробка різних типів зображень і документів, що дозволяє застосовувати його як для роботи з вебконтентом, так і для підготовки матеріалів до публікації чи передачі мережею. Завдяки інтуїтивно зрозумілому підходу до взаємодії з користувачем процес оптимізації не потребує складних налаштувань і може виконуватися з мінімальними витратами часу [9], [11], [15].

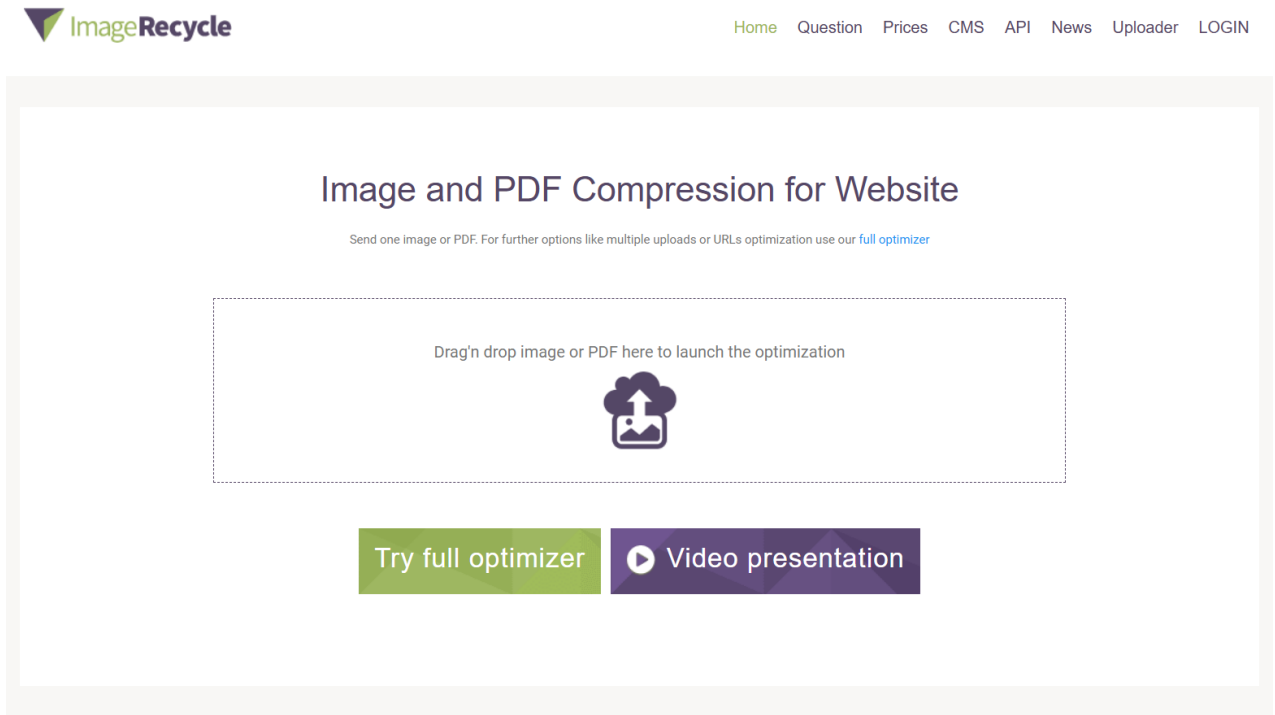


Рисунок 1.2 – Програмне забезпечення ImageRecycle [15]

Особливістю функціонування є підтримка механізму безпосереднього додавання файлів до середовища обробки, що реалізується через зручну взаємодію з інтерфейсом. Такий підхід дозволяє швидко передавати зображення до системи та одразу отримувати результат без необхідності виконання додаткових дій. У поєднанні з автоматизованими алгоритмами стиснення це формує безперервний і зрозумілий робочий процес, який підходить як для початківців, так і для досвідчених користувачів [9], [11], [15].

Окрему увагу у програмному забезпеченні ImageRecycle приділено можливості аналізу вебресурсів, у межах якого формується звіт щодо ефективності використання зображень. У результаті такого аналізу визначаються елементи, які можуть бути оптимізовані з метою підвищення швидкості завантаження сторінок. Це дозволяє не лише виконувати стиснення файлів, а й оцінювати їх вплив на загальну продуктивність вебсайту, що є важливим аспектом сучасної веброботи [9], [11], [15].

Функціональні можливості програмного засобу також охоплюють автоматичну зміну параметрів зображень під час обробки, що дозволяє

адаптувати їх до вимог конкретного середовища використання. Додатково реалізовано механізми оптимізації завантаження, які сприяють поступовому відображенню графічного контенту та зменшенню навантаження на систему під час відкриття сторінок. Використання розподілених мереж доставки даних дозволяє забезпечити швидший доступ до зображень незалежно від розташування користувача [9], [11], [15].

Важливою перевагою програмного забезпечення ImageRecycle є інтеграція з популярними системами керування контентом, що дозволяє виконувати оптимізацію безпосередньо в межах використовуваної платформи. Це усуває необхідність переходу між різними інструментами та спрощує загальний процес роботи з графічними матеріалами. У результаті забезпечується безперервність обробки та підвищується ефективність управління вмістом [9], [11], [15].

Таким чином, програмне забезпечення ImageRecycle поєднує простоту використання, автоматизацію процесів і можливість аналітичної оцінки, що робить його зручним інструментом для перетворення та стиснення зображень. Його застосування сприяє покращенню продуктивності вебресурсів і оптимізації використання графічних даних без суттєвого ускладнення робочого процесу [9], [11], [15].

Програмне забезпечення ImageRecycle має такі особливості [9], [11], [15]:

- універсальна підтримка форматів: у програмному забезпеченні ImageRecycle забезпечується обробка різних типів графічних файлів і документів, що дозволяє використовувати інструмент у широкому спектрі задач веброзробки [9], [11], [15];

- зручний інтерфейс взаємодії: у програмному забезпеченні ImageRecycle реалізовано простий механізм додавання файлів, що значно спрощує процес оптимізації зображень навіть для не зовсім досвідчених користувачів [9], [11], [15];

- автоматизація обробки: у програмному забезпеченні ImageRecycle виконання стиснення та зміни параметрів зображень здійснюється без необхідності ручного налаштування кожного етапу [9], [11], [15];
- аналіз вебресурсів: у програмному забезпеченні ImageRecycle передбачено формування звітів, що дозволяють оцінити ефективність використання зображень та визначити потенціал їх оптимізації [9], [11], [15];
- інтеграція з CMS: у програмному забезпеченні ImageRecycle підтримується взаємодія з популярними системами керування контентом, що забезпечує зручність використання у межах вебплатформ [9], [11], [15];
- оптимізація швидкодії: зменшення розміру графічних файлів сприяє прискоренню завантаження сторінок та покращенню загальної продуктивності вебресурсу [9], [11], [15];
- адаптивне налаштування параметрів: у програмному забезпеченні ImageRecycle обробка зображень виконується з урахуванням їх характеристик, що дозволяє досягти балансу між якістю та обсягом даних [9], [11], [15];
- підтримка відкладеного завантаження: у програмному забезпеченні ImageRecycle реалізовано механізм, що дозволяє завантажувати зображення поступово, зменшуючи навантаження на систему [9], [11], [15];
- використання мереж доставки контенту: у програмному забезпеченні ImageRecycle забезпечується швидке передавання зображень користувачам незалежно від їх географічного розташування [9], [11], [15];
- масштабованість використання: програмний засіб ImageRecycle придатний як для невеликих проєктів, так і для ресурсів із великим обсягом графічного контенту [9], [11], [15].

Серед недоліків програмного забезпечення ImageRecycle можна відзначити залежність від зовнішньої інфраструктури, оскільки обробка виконується не локально, що може впливати на швидкість роботи та вимагати стабільного підключення до мережі. Крім того, безкоштовні можливості є обмеженими, що знижує ефективність використання без додаткових витрат, а також можливі обмеження щодо обсягу оброблюваних даних. Окремо варто

враховувати, що передача файлів до стороннього сервісу може викликати питання щодо конфіденційності даних, а інтеграція з різними платформами інколи потребує додаткового налаштування [9], [11], [15].

Програмне забезпечення Zamzar (рис. 1.3) є потужним онлайн інструментом для перетворення та стиснення зображень, яке дозволяє працювати з великою кількістю графічних форматів. Воно надає можливість завантажувати файли безпосередньо з комп'ютера або вказувати URL-адреси для обробки, після чого користувач отримує результат через посилання на сторінці завантаження або на електронну пошту. Програмне забезпечення підтримує масову конвертацію, що робить його зручним для швидкого оброблення багатьох файлів одночасно [12], [14], [16].

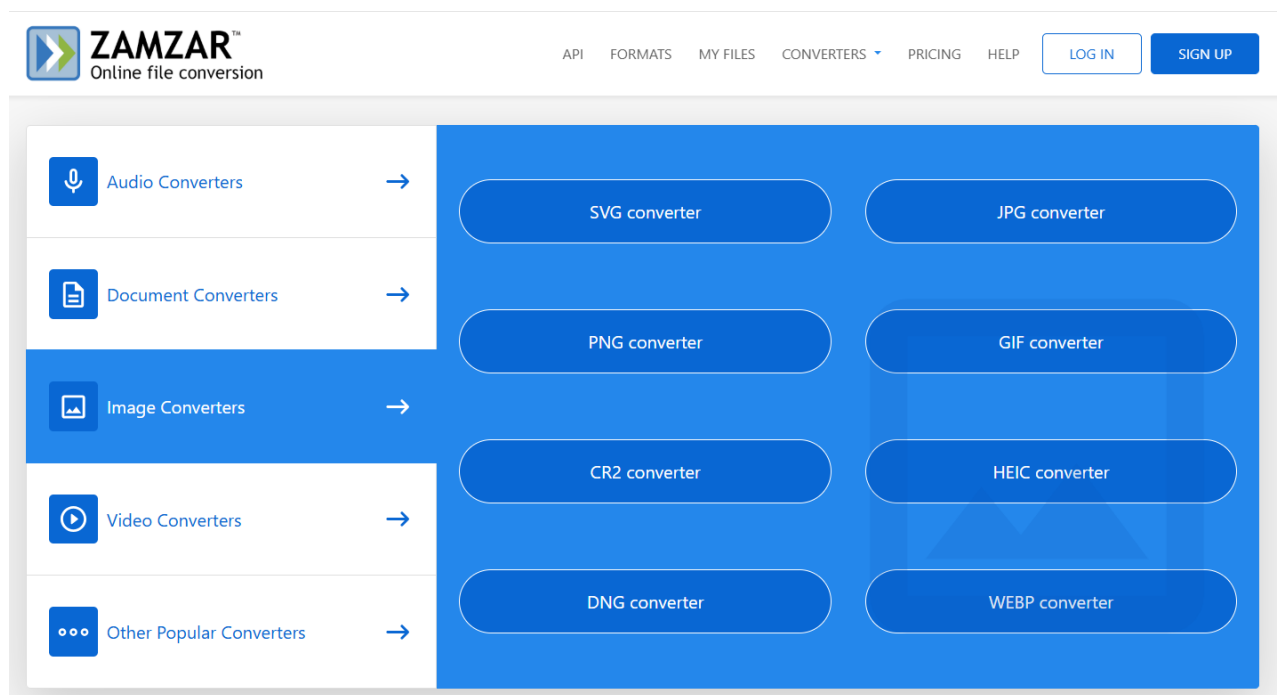


Рисунок 1.3 – Програмне забезпечення Zamzar [16]

Функціонування програмного забезпечення Zamzar передбачає автоматичне стиснення зображень із можливістю регулювання співвідношення якості та розміру файлу. Платна версія надає пріоритетну обробку і зменшує обмеження щодо швидкості та кількості конвертацій, тоді як безкоштовні користувачі мають певні ліміти. Інструмент також пропонує

зручний API для інтеграції з десктопними або веб-застосунками, що дозволяє виконувати конвертацію файлів безпосередньо з програмного коду, використовуючи прості та зрозумілі команди [12], [14], [16].

Особливістю програмного забезпечення Zamzar є його універсальність: крім популярних графічних форматів, підтримуються документи, відео, аудіо, CAD-файли та електронні книги, що робить його багатфункціональним рішенням для роботи з різними типами файлів. Система забезпечує просту та зрозумілу документацію, приклади коду і підтримку для розробників, що спрощує інтеграцію та автоматизацію процесів перетворення і стиснення даних [12], [14], [16].

Інтерфейс програмного забезпечення Zamzar реалізований як онлайн платформа, де користувач може швидко додавати файли та відслідковувати стан конвертації. Механізм пакетної обробки та управління кредитами дозволяє ефективно планувати обсяг обробки і оптимізувати витрати для користувачів із великими обсягами даних. Завдяки таким рішенням забезпечується баланс між простотою використання та функціональністю, що робить інструмент надійним для як епізодичного, так і системного використання в роботі з графікою [12], [14], [16].

Таким чином, програмне забезпечення Zamzar пропонує комплексне середовище для перетворення та стискання зображень, поєднуючи зручність онлайн-доступу, підтримку широкого спектру форматів, автоматизацію та можливість інтеграції з програмними продуктами, що значно спрощує роботу з графічними даними і підвищує ефективність обробки файлів [12], [14], [16].

Програмне забезпечення Zamzar має такі особливості [12], [14], [16]:

- універсальна підтримка форматів: забезпечується конвертація широкого спектру графічних, документальних, аудіо- та відеофайлів, що дозволяє використовувати інструмент для різних типів даних [12], [14], [16];

- онлайн-доступ: програмне забезпечення Zamzar працює без встановлення на комп'ютері, надаючи можливість обробки файлів через веббраузер [12], [14], [16];

- масова обробка файлів: у програмному забезпеченні Zamzar підтримується пакетне завантаження та обробка великої кількості файлів одночасно, що оптимізує час користувача [12], [14], [16];
- інтегрований API: у програмному забезпеченні Zamzar передбачена можливість автоматизованої конвертації з програмного коду, що полегшує інтеграцію з іншими застосунками [12], [14], [16];
- регулювання параметрів стиснення: користувачі програмного забезпечення Zamzar отримують контроль над співвідношенням якості та розміру файлу, забезпечуючи баланс між швидкістю завантаження та візуальними характеристиками [12], [14], [16];
- підтримка пакетних кредитів: у програмному забезпеченні Zamzar передбачено систему планів і кредитів для користувачів із великими обсягами роботи, що дозволяє економити ресурси [12], [14], [16];
- прозора документація: надані зрозумілі інструкції та приклади коду для швидкого початку роботи з конвертацією [12], [14], [16];
- контроль якості: платні користувачі програмного забезпечення Zamzar отримують пріоритетну обробку та можливість уникнути затримок у виконанні завдань [12], [14], [16];
- зручність використання: інтерфейс програмного забезпечення Zamzar орієнтований на швидке додавання файлів, перегляд стану обробки та завантаження результатів [12], [14], [16];
- гнучкість інтеграції: програмне забезпечення Zamzar сумісне з різними операційними системами та платформами, що робить його універсальним рішенням [12], [14], [16].

Серед недоліків програмного забезпечення Zamzar можна відзначити обмеження безкоштовної версії щодо розміру файлів та кількості конвертацій на день, а також необхідність очікувати посилення для завантаження оброблених файлів, що інколи може знижувати оперативність роботи користувача [12], [14], [16].

Порівняльну характеристику програмного забезпечення для перетворення та стиснення зображень наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для перетворення та стиснення зображень

Критерій порівняння	EWWW Image Optimization [14]	ImageRecycle [15]	Zamzar [16]
підтримка локальної обробки зображень	+	+—	—
робота без встановлення	+—	+	+
інтеграція з системами управління вмістом	+	+	+—
можливість API-інтеграції	+—	+—	+
контроль якості стиснення	+	+	+—
простота інтерфейсу	+—	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для перетворення та стиснення зображень. Недоліки існуючого програмного забезпечення для перетворення та стиснення зображень проявляються у низці аспектів, пов'язаних як з технічними обмеженнями алгоритмів, так і з особливостями їх практичної реалізації. У процесі обробки графічних даних часто спостерігається компроміс між ступенем стиснення та якістю зображення, внаслідок чого при інтенсивному зменшенні обсягу файлу виникають втрати деталей, поява артефактів і спотворення кольорів. Це особливо критично у випадках, коли зображення використовуються для професійних або наукових цілей, де точність відображення має принципове значення. Суттєвим недоліком є також залежність ефективності стиснення від типу зображення та

обраного формату. Алгоритми, що добре працюють для фотографій, можуть виявитися неефективними для графіки з чіткими контурами або текстом, і навпаки. У результаті виникає необхідність додаткового налаштування параметрів або використання різних інструментів для різних задач, що ускладнює процес автоматизації та знижує універсальність програмного забезпечення. Не менш важливим є питання сумісності та стандартизації. Існування великої кількості форматів і варіацій їх реалізації призводить до ситуацій, коли певні програми некоректно обробляють окремі файли або не підтримують новітні стандарти. Це створює додаткові труднощі при інтеграції різних програмних систем та обміні даними між ними. Тому актуальною є розробка програмного забезпечення для перетворення та стиснення зображень.

При розробці програмного забезпечення для перетворення та стиснення зображень необхідно забезпечити такі функціональні вимоги:

- підтримка можливості завантаження зображень у різних форматах у програмне забезпечення;
- підтримка можливості вибору методів перетворення та стиснення зображень, таких як вейвлети Хаара та Добеши;
- можливість перегляду оригінального та обробленого зображення;
- підтримка відображення статусу процесу обробки зображення;
- підтримка можливості збереження оброблених зображень у вибраному форматі та шляхом користувача;
- підтримка ведення журналів дій користувача та обробки зображень для подальшого аналізу та відстеження помилок.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для перетворення та стиснення зображень являє собою клас програмних систем, призначених для обробки графічних файлів з метою зміни їх формату, розміру або зменшення обсягу

зберігання без значної втрати якості. Такі програмні системи дозволяють оптимізувати зображення для подальшого використання у вебсайтах, мультимедійних проєктах або базах даних, підвищуючи швидкість завантаження та економлячи ресурси сховища. В їх основі лежать алгоритми обчислювальної обробки, що аналізують піксельні дані, визначають повторювані або непотрібні елементи та ефективно перетворюють інформацію у стислий формат.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для перетворення та стиснення зображень. Визначено, що недоліки існуючого програмного забезпечення для перетворення та стиснення зображень проявляються у низці аспектів, пов'язаних як з технічними обмеженнями алгоритмів, так і з особливостями їх практичної реалізації. У процесі обробки графічних даних часто спостерігається компроміс між ступенем стиснення та якістю зображення, внаслідок чого при інтенсивному зменшенні обсягу файлу виникають втрати деталей, поява артефактів і спотворення кольорів. Це особливо критично у випадках, коли зображення використовуються для професійних або наукових цілей, де точність відображення має принципове значення. Суттєвим недоліком є також залежність ефективності стиснення від типу зображення та обраного формату. Алгоритми, що добре працюють для фотографій, можуть виявитися неефективними для графіки з чіткими контурами або текстом, і навпаки. У результаті виникає необхідність додаткового налаштування параметрів або використання різних інструментів для різних задач, що ускладнює процес автоматизації та знижує універсальність програмного забезпечення. Не менш важливим є питання сумісності та стандартизації. Існування великої кількості форматів і варіацій їх реалізації призводить до ситуацій, коли певні програми некоректно обробляють окремі файли або не підтримують новітні стандарти. Це створює додаткові труднощі при інтеграції різних програмних систем та обміні даними між ними. Тому актуальною є

розробка програмного забезпечення для перетворення та стиснення зображень.

Сформульовано функціональні вимоги до програмного забезпечення для перетворення та стиснення зображень.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для перетворення та стиснення зображень було використано мову програмування C# [17], [18].

Мова програмування C# є однією з ключових складових сучасної екосистеми розробки програмного забезпечення, що поєднує у собі риси об'єктно-орієнтованого, компонентного та частково функціонального підходів до програмування. У межах платформи .NET вона еволюціонувала від інструменту для створення переважно настільних застосунків до універсального засобу розробки, придатного для реалізації широкого спектра програмних рішень, від вебсервісів до високонавантажених розподілених систем [17], [18].

Суттєвою характеристикою мови програмування C# є високий рівень типізації, що сприяє підвищенню надійності програмного коду та дозволяє виявляти значну частину помилок ще на етапі компіляції. Поряд із цим активно розвиваються механізми, спрямовані на підвищення виразності мови, зокрема впровадження лямбда-виразів, шаблонів, інтегрованих запитів до даних та розширених можливостей роботи з асинхронністю. Такі особливості дозволяють формувати більш компактний і зрозумілий код, що має важливе значення у складних програмних системах [17], [18].

У контексті сучасних підходів до розробки особливу увагу привертає підтримка асинхронного програмування. Це дає змогу ефективно працювати з операціями введення-виведення, мережевими запитами та обробкою великих обсягів даних без блокування основного потоку виконання. У результаті підвищується продуктивність застосунків і покращується взаємодія з користувачем [17], [18].

Важливим аспектом є також багатоплатформеність цієї мови. Завдяки розвитку платформи виконання, програмні рішення можуть функціонувати не лише у середовищі операційних систем родини Windows, але й на інших

платформах, включаючи Linux та macOS. Це розширює сферу застосування мови та робить її конкурентоспроможною у порівнянні з іншими сучасними технологіями [17], [18].

Вбудовані механізми мови програмування C# для контролю типів, керування пам'яттю та виконання коду у віртуальному середовищі сприяють зниженню ризиків виникнення критичних помилок і вразливостей. Це особливо актуально при створенні систем, що працюють з важливими або конфіденційними даними [17], [18].

Крім того, мова характеризується високим рівнем підтримки з боку спільноти розробників та наявністю значної кількості інструментів для розробки, тестування й супроводу програмного забезпечення. Інтегровані середовища розробки, такі як Visual Studio, забезпечують розширені можливості налагодження, профілювання та автоматизації процесів, що сприяє підвищенню якості кінцевого продукту [17], [18].

Таким чином, зазначена мова програмування постає як зрілий, гнучкий і потужний інструмент, здатний забезпечити ефективну реалізацію складних програмних систем, відповідаючи сучасним вимогам до продуктивності, масштабованості та зручності супроводу програмного коду [17], [18].

Доцільність вибору мови програмування C# для розробки програмного забезпечення, орієнтованого на перетворення та стиснення зображень, обумовлюється поєднанням високого рівня абстракції, продуктивності та широких можливостей інтеграції з сучасними технологічними платформами. У середовищі .NET забезпечується доступ до розвиненої екосистеми бібліотек і засобів, що спрощують реалізацію складних алгоритмів обробки графічних даних і водночас дозволяють зосередитися на логіці застосунку, а не на низькорівневих деталях керування пам'яттю чи потоками виконання [17], [18].

Важливим фактором є те, що в межах цієї технологічної платформи реалізовано ефективні механізми роботи з пам'яттю, включаючи автоматичне керування її виділенням і звільненням. Це знижує ймовірність виникнення помилок, пов'язаних із витоками пам'яті, які є критичними при обробці

великих масивів зображень. Одночасно сучасні версії середовища виконання забезпечують оптимізацію продуктивності, що дозволяє досягати достатньо високої швидкості обробки навіть при використанні складних алгоритмів стиснення [17], [18].

Значна перевага мови програмування C# полягає у наявності вбудованих і сторонніх бібліотек для роботи з графікою, зокрема таких як System.Drawing або ImageSharp, які надають інструменти для маніпулювання пікселями, зміни форматів, масштабування та виконання інших операцій над зображеннями. Це дозволяє істотно скоротити час розробки та підвищити надійність програмного забезпечення за рахунок використання перевірених компонентів [17], [18].

Окрім цього, важливою перевагою мови програмування C# є підтримка багатопоточності та асинхронного програмування, що надає можливість ефективно обробляти великі обсяги графічних даних паралельно. У задачах стиснення та конвертації зображень це має особливе значення, оскільки дозволяє підвищити пропускну здатність системи та скоротити час виконання операцій без суттєвого ускладнення коду [17], [18].

Необхідно також враховувати високий рівень інтеграції з операційними системами та можливість створення як настільних, так і веб- чи хмарних застосунків. Завдяки цьому програмні рішення можуть бути легко адаптовані до різних сценаріїв використання, включаючи обробку зображень на сервері, у клієнтських застосунках або в розподілених середовищах [17], [18].

Таким чином, використання зазначеної мови програмування забезпечує збалансоване поєднання зручності розробки, продуктивності та масштабованості, що робить її обґрунтованим вибором для створення ефективних систем перетворення та стиснення зображень у сучасних умовах розвитку інформаційних технологій [17], [18].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для перетворення та стиснення зображень наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для перетворення та стиснення зображень

Критерій порівняння мов програмування	Мова програмування		
	C#	C++	Java
Продуктивність обробки зображень	+	+	+—
Швидкість розробки	+	—	+—
Зручність роботи з пам'яттю	+	—	+
Якість інструментів розробки	+	+—	+
Баланс продуктивності та простоти	+	—	+—

Отже, для реалізації програмного забезпечення для перетворення та стиснення зображень обрано мову програмування C#, що обумовлюється поєднанням високого рівня абстракції, продуктивності та широких можливостей інтеграції з сучасними технологічними платформами. Крім того, у середовищі .NET забезпечується доступ до розвиненої екосистеми бібліотек і засобів, що спрощують реалізацію складних алгоритмів обробки графічних даних і водночас дозволяють зосередитися на логіці застосунку, а не на низькорівневих деталях керування пам'яттю чи потоками виконання.

2.2 Вибір середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень

В якості середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень було обрано середовище Visual Studio [19], [20].

Середовище розробки Visual Studio доцільно розглядати не лише як інструмент для написання коду, але як комплексну платформу, що формує

цілісну інфраструктуру створення програмних продуктів різної складності. У його межах реалізується концепція інтегрованого підходу до розробки, коли всі ключові етапи, проєктування, реалізація, тестування та супровід, об'єднані в єдиному середовищі, що сприяє підвищенню узгодженості та керованості процесу [19], [20].

Значною особливістю середовища розробки Visual Studio є гнучкість налаштування, яка дозволяє адаптувати середовище під конкретні вимоги проєкту або індивідуальні вподобання розробника. Підтримка розширень і модулів забезпечує можливість інтеграції додаткових інструментів, що розширюють функціональність та дозволяють працювати з різними технологіями без необхідності зміни основного середовища. У цьому контексті формується екосистема, яка здатна еволюціонувати разом із потребами розробки [19], [20].

Окрему увагу привертає підтримка сучасних підходів до організації програмного коду, зокрема використання шаблонів проєктування, автоматичної генерації структур проєктів та засобів рефакторингу. Це дозволяє підтримувати високий рівень якості коду, зменшувати його складність і забезпечувати можливість подальшого масштабування програмних рішень без суттєвих витрат ресурсів [19], [20].

У межах платформи .NET забезпечується тісна інтеграція з різноманітними бібліотеками та сервісами, що відкриває широкі можливості для створення як локальних, так і розподілених систем. При цьому реалізуються механізми автоматичного складання проєктів, керування залежностями та організації середовищ виконання, що спрощує процес розгортання програмного забезпечення [19], [20].

Важливим аспектом є також підтримка тестування на різних рівнях, що дозволяє здійснювати перевірку як окремих компонентів, так і всієї системи в цілому. Інтегровані засоби тестування сприяють ранньому виявленню помилок та підвищують надійність програмного забезпечення ще до етапу його впровадження [19], [20].

Не менш значущою є орієнтація на продуктивність розробника, яка досягається за рахунок автоматизації рутинних операцій, зручної навігації по коду та можливості швидкого переходу між різними частинами проєкту. Це дозволяє зосередитися на вирішенні складних задач, мінімізуючи витрати часу на допоміжні дії [19], [20].

Таким чином, середовища розробки Visual Studio постає як потужний інструмент, що поєднує у собі засоби проєктування, реалізації та супроводу програмних систем, забезпечуючи високий рівень ефективності розробки та створюючи сприятливі умови для побудови якісних і масштабованих програмних рішень [19], [20].

Доцільність використання середовища розробки Visual Studio для створення програмного забезпечення, призначеного для перетворення та стиснення зображень, визначається високим рівнем інтеграції інструментів, що охоплюють повний життєвий цикл розробки — від написання коду до тестування та оптимізації продуктивності. У межах платформи .NET забезпечується тісна взаємодія між мовою програмування, бібліотеками та засобами налагодження, що дозволяє формувати цілісне середовище для ефективної реалізації складних алгоритмів обробки зображень [19], [20].

У процесі розробки особливого значення набуває можливість глибокого аналізу виконання програм, оскільки операції стиснення та перетворення графічних даних часто є ресурсоємними. Вбудовані засоби профілювання середовища розробки Visual Studio дозволяють виявляти вузькі місця у програмному коді, оцінювати використання пам'яті та процесорного часу, а також оптимізувати алгоритми без необхідності залучення сторонніх інструментів. Це сприяє підвищенню ефективності кінцевого програмного продукту [19], [20].

Не менш важливою особливістю середовища розробки Visual Studio є підтримка інтелектуальних механізмів редагування коду, які забезпечують автоматичне доповнення, перевірку синтаксису та аналіз помилок у режимі реального часу. Такий підхід знижує ймовірність виникнення логічних і

синтаксичних помилок, що особливо актуально при реалізації складних математичних і алгоритмічних операцій, характерних для обробки зображень. Одночасно підвищується швидкість розробки та спрощується супровід програмного коду [19], [20].

Значна увага у середовищі розробки Visual Studio приділяється можливостям налагодження, які дозволяють детально відстежувати виконання програми, аналізувати значення змінних та перевіряти коректність роботи окремих модулів. У задачах, пов'язаних із графічними даними, це має особливе значення, оскільки навіть незначні помилки можуть призводити до помітних спотворень результатів обробки [19], [20].

Крім того, середовище забезпечує зручні механізми інтеграції з системами контролю версій, що дозволяє організувати колективну розробку та забезпечити контроль змін у проєкті. Це є важливим фактором у випадках, коли програмне забезпечення розробляється у команді або потребує тривалого супроводу та вдосконалення [19], [20].

Важливо також враховувати можливість створення різних типів застосунків у межах середовища розробки Visual Studio, що дозволяє реалізовувати як настільні програми для локальної обробки зображень, так і серверні рішення для обробки даних у хмарних середовищах. Така універсальність сприяє розширенню сфер застосування розробленого програмного забезпечення [19], [20].

Таким чином, використання середовища розробки Visual Studio забезпечує високий рівень автоматизації процесів створення програм, підвищує якість коду та дозволяє ефективно реалізовувати складні алгоритми обробки графічної інформації, що обґрунтовує його вибір для задач перетворення та стиснення зображень у сучасних умовах розвитку інформаційних технологій [19], [20].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	JetBrains Rider	MonoDevelop
Інструменти налагодження	+	+	+—
Засоби профілювання продуктивності	+	+—	—
Інтеграція з графічними бібліотеками	+	+	+—
Швидкість роботи середовища	+	+—	+—
Зручність інтерфейсу	+	+—	+—

Таким чином, для створення програмного забезпечення для перетворення та стиснення зображень обрано середовище розробки Visual Studio, що забезпечує високий рівень автоматизації процесів створення програм, підвищує якість коду та дозволяє ефективно реалізовувати складні алгоритми обробки графічної інформації. Крім того, середовище Visual Studio надає зручні засоби налагодження програм, які дозволяють детально відстежувати виконання програми, аналізувати значення змінних та перевіряти коректність роботи окремих модулів.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для перетворення та стиснення зображень обрано мову програмування C#, що обумовлюється поєднанням високого рівня абстракції, продуктивності та широких

можливостей інтеграції з сучасними технологічними платформами. Крім того, у середовищі .NET забезпечується доступ до розвиненої екосистеми бібліотек і засобів, що спрощують реалізацію складних алгоритмів обробки графічних даних і водночас дозволяють зосередитися на логіці застосунку, а не на низькорівневих деталях керування пам'яттю чи потоками виконання.

Для створення програмного забезпечення для перетворення та стиснення зображень обрано середовище розробки Visual Studio, що забезпечує високий рівень автоматизації процесів створення програм, підвищує якість коду та дозволяє ефективно реалізовувати складні алгоритми обробки графічної інформації. Крім того, середовище Visual Studio надає зручні засоби налагодження програм, які дозволяють детально відстежувати виконання програми, аналізувати значення змінних та перевіряти коректність роботи окремих модулів.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для перетворення та стиснення зображень

Структуру програмного забезпечення для перетворення та стиснення зображень наведено на рис. 3.1.

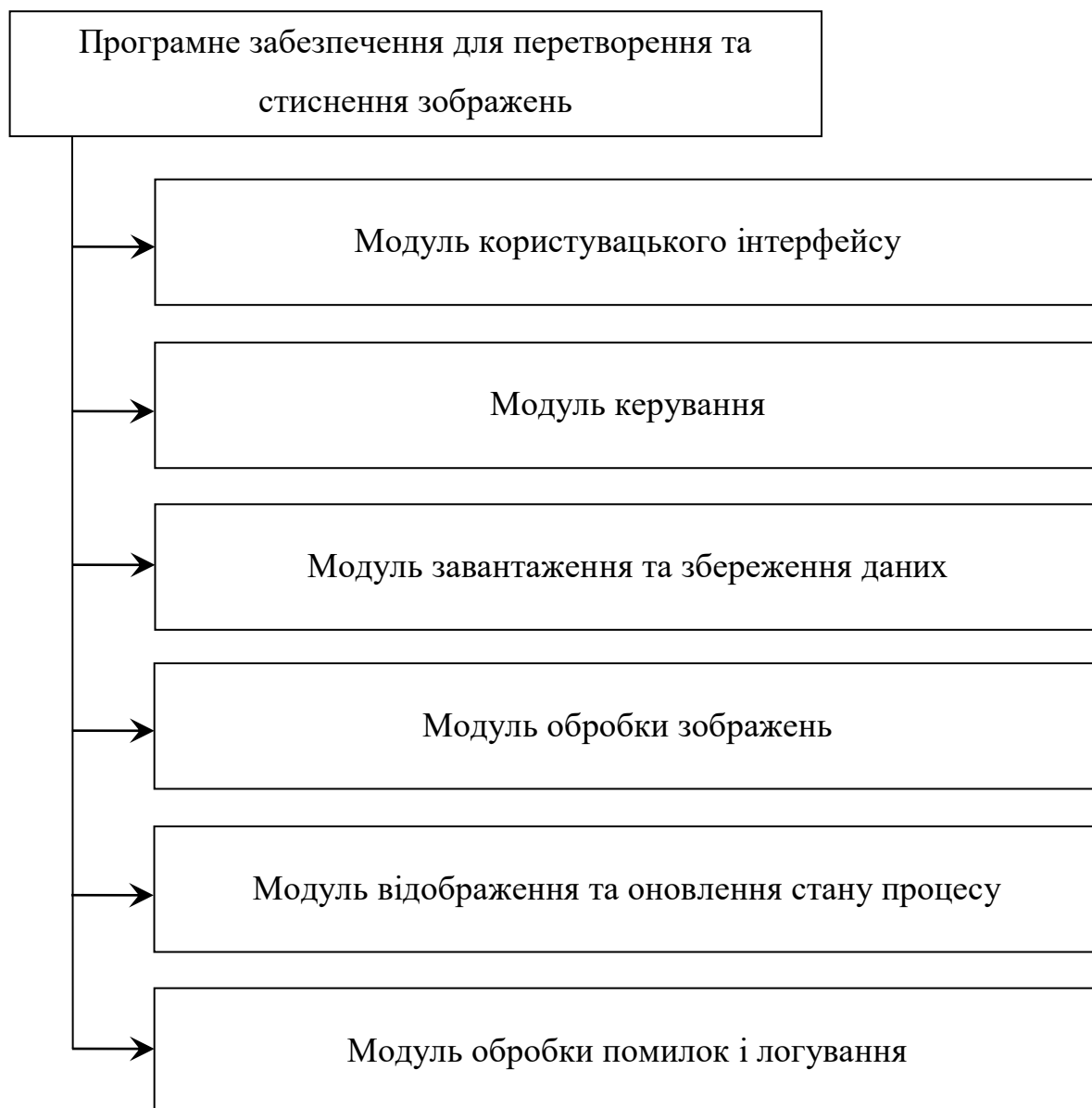


Рисунок 3.1 – Структура програмного забезпечення для перетворення та стиснення зображень

Структура програмного забезпечення для перетворення та стиснення зображень у вигляді настільного застосунку для операційної системи Windows являє собою сукупність взаємопов'язаних функціональних модулів, кожен з яких відповідає за окремий аспект обробки даних і взаємодії з користувачем. До складу ПЗ входять такі модулі: модуль користувацького інтерфейсу, модуль керування, модуль завантаження та збереження даних, модуль обробки зображень, модуль відображення та оновлення стану процесу, модуль обробки помилок і логування.

Модуль користувацького інтерфейсу забезпечує взаємодію з користувачем через графічні елементи віконного середовища. У межах цього модуля реалізується можливість вибору файлу зображення, відображення завантаженого графічного матеріалу у відповідній області, а також виведення інформації про поточний стан обробки. Цей компонент відповідає за візуалізацію результатів і забезпечення зручності використання застосунку.

Окреме місце займає модуль керування, який виконує функції координації роботи всієї системи. Він приймає запити від інтерфейсу, ініціює процеси перетворення та стиснення, визначає послідовність виконання операцій і забезпечує передачу даних між іншими модулями. У цьому модулі реалізується логіка керування станами програми, включаючи початок, виконання та завершення обробки.

Важливим компонентом є модуль завантаження та збереження даних, який відповідає за роботу з файловою системою. Його функціональність охоплює відкриття графічних файлів різних форматів, перетворення їх у внутрішнє представлення, а також запис результатів обробки у файл. Даний модуль забезпечує коректність читання і запису даних, а також обробку можливих помилок, пов'язаних із доступом до файлів.

Центральним елементом програмної системи виступає модуль обробки зображень, у якому реалізуються алгоритми перетворення та стиснення. У межах цього компонента здійснюється аналіз піксельних даних, виконуються математичні операції та формуються результати, що передаються до інших

частин системи. Тут відбувається основна обчислювальна робота, що визначає ефективність і якість функціонування програмного забезпечення.

Модуль відображення та оновлення стану процесу забезпечує інформування користувача про хід виконання операцій. У ньому реалізується механізм передачі повідомлень про поточний етап обробки, рівень завершеності задачі або можливі помилки. Це дозволяє підвищити прозорість роботи програми та забезпечити контроль над її виконанням.

Модуль обробки помилок і логування відповідає за фіксацію подій, що виникають у процесі роботи застосунку, а також за обробку виняткових ситуацій. Його функціонування сприяє підвищенню надійності системи, оскільки дозволяє виявляти причини збоїв і забезпечує можливість їх подальшого аналізу.

Запропонована структура програмного забезпечення для перетворення та стиснення зображень характеризується модульністю, чітким розподілом функцій і орієнтацією на зручність використання та ефективність обробки зображень. Такий підхід забезпечує гнучкість у розвитку системи, спрощує її модернізацію та дозволяє досягти високого рівня якості кінцевого програмного продукту.

3.2 Функціонування програмного забезпечення для перетворення та стиснення зображень

Функціонування програмного забезпечення для перетворення та стиснення зображень подамо за допомогою схеми, зображеної на рис. 3.2. Послідовність функціонування програмного забезпечення для перетворення та стиснення зображень подано у вигляді впорядкованого ланцюга дій, що відображає логіку переходу між станами системи.



Рисунок 3.2 – Схема функціонування програмного забезпечення для перетворення та стиснення зображень

Початковим етапом виступає запуск застосунку, у результаті чого відбувається ініціалізація основних компонентів і відображення головного вікна інтерфейсу. Після цього система переходить у стан очікування дій користувача. Наступною дією є ініціювання процесу вибору файлу, під час якого користувач відкриває діалогове вікно та обирає зображення з файлової системи. Далі виконується перевірка коректності обраного файлу, що включає визначення формату, доступності та можливості подальшої обробки. У разі невідповідності вимогам формується повідомлення про помилку та здійснюється повернення до стану очікування.

У випадку успішної перевірки відбувається завантаження зображення у внутрішнє представлення програми з подальшим відображенням у відповідній області інтерфейсу. Після цього здійснюється вибір параметрів обробки, що включає визначення способу стиснення та, за необхідності, додаткових

налаштувань. На основі отриманих параметрів формується команда на запуск процесу обробки.

Далі програмне забезпечення переходить до етапу виконання обчислювальних операцій, під час якого здійснюється перетворення зображення та його стиснення відповідно до обраного методу. У процесі виконання забезпечується оновлення статусу, що відображається у спеціальній області інтерфейсу, де фіксується прогрес виконання або поточний стан операції. Після завершення обробки формується результат, який передається до модуля відображення.

Наступною дією є виведення обробленого зображення та оновлення інтерфейсу із відображенням завершеного статусу. Після цього користувачу надається можливість збереження результату, що супроводжується вибором місця розташування та формату вихідного файлу. У разі успішного збереження система фіксує завершення операції та повертається у стан очікування нових дій.

Завершальним етапом є закриття застосунку, яке може бути ініційоване користувачем у будь-який момент, із попереднім завершенням активних процесів та коректним звільненням ресурсів.

Таким чином, функціонування програмного забезпечення характеризується чітко визначеною послідовністю дій, що забезпечує логічну цілісність процесу обробки зображень, передбачає перевірку вхідних даних, виконання обчислювальних операцій та надання користувачу результатів у зручній формі, що робить систему зрозумілою для моделювання у вигляді блок-схеми та ефективною у практичному використанні.

3.3 Реалізація алгоритмів обробки зображень

У межах програмного забезпечення обробка зображень реалізується як послідовність обчислювальних процедур, що забезпечують перетворення вхідних графічних даних у форму, придатну для ефективного стиснення та

подальшого збереження. Організація обчислень базується на поетапному підході, за якого кожна операція виконує чітко визначену функцію та передає результати наступному етапу обробки. Така побудова дозволяє забезпечити контрольованість процесу, підвищити точність обчислень і спростити налагодження програмної системи.

На початковому етапі здійснюється представлення зображення у внутрішньому форматі, що використовується для подальших обчислень. Графічні дані інтерпретуються як двовимірний масив пікселів, де кожен елемент відповідає певній точці зображення та містить інформацію про її кольорні характеристики. У випадку кольорових зображень використовується багатовимірне подання, в якому кожен піксель складається з кількох компонентів, що відповідають окремим каналам. Для підвищення ефективності обчислень може застосовуватися перетворення до одноканального представлення або нормалізація значень інтенсивності, що дозволяє уніфікувати подальші операції.

Після формування внутрішньої структури даних виконується підготовчий етап, який передбачає приведення зображення до вигляду, зручного для застосування математичних перетворень, зокрема, за необхідності виконується зміна розмірності масиву, вирівнювання меж або доповнення даних до необхідного формату. Такі дії спрямовані на забезпечення коректності виконання алгоритмів, що використовуються на наступних етапах.

Ключовим елементом обробки є застосування вейвлет-перетворення, яке дозволяє представити зображення у вигляді сукупності коефіцієнтів, що відображають як глобальні, так і локальні особливості структури даних. У процесі реалізації цього підходу обчислення організовуються як послідовне застосування фільтрації до рядків і стовпців зображення. У результаті формується набір піддіапазонів, кожен з яких містить інформацію про різні частотні складові зображення. Такий підхід дозволяє виділити основні

структурні елементи та відокремити деталі, що мають менший вплив на візуальне сприйняття.

Отримані коефіцієнти піддаються подальшій обробці, що спрямована на зменшення обсягу даних. У цьому контексті виконується аналіз значущості коефіцієнтів і відкидання або модифікація тих із них, які мають незначний внесок у загальну структуру зображення. Завдяки цьому досягається ефект стиснення, оскільки кількість інформації, необхідної для збереження, істотно зменшується. Водночас забезпечується збереження основних характеристик зображення, що дозволяє відтворити його з прийнятною якістю.

На завершальному етапі формується вихідне представлення даних, яке може бути або збережене у файл, або використане для подальшого відображення у графічному інтерфейсі. У випадку необхідності виконується зворотне перетворення, що дозволяє відновити зображення з отриманих коефіцієнтів. Цей процес також організовується як послідовність обчислювальних операцій, які виконуються у зворотному порядку до початкового перетворення.

Схему реалізації алгоритмів обробки зображень наведено на рис. 3.3, що відображає узагальнений процес обробки зображення у програмному забезпеченні та демонструє логіку переходу від початкових даних до кінцевого результату. Початковим етапом є отримання вхідного зображення у вигляді файлу, після чого воно переводиться у внутрішнє представлення, що дозволяє виконувати обчислення на рівні пікселів. Такий підхід забезпечує зручність подальших математичних операцій, оскільки дані подаються у вигляді структурованої матриці. Далі виконується попередня обробка, у межах якої здійснюється підготовка зображення до застосування перетворень. Після цього ключовим етапом є вейвлет-перетворення, яке забезпечує перехід від просторового представлення до частотного. У результаті формується набір коефіцієнтів, що відображають різні рівні деталізації зображення.



Рисунок 3.3 – Схема реалізації алгоритмів обробки зображень

Наступний блок відповідає за обробку цих коефіцієнтів, у ході якої відбувається зменшення обсягу даних за рахунок усунення або спрощення малозначущих компонентів. Завершальний етап передбачає формування результату, який може бути або збережений у файл, або використаний для подальшого відображення. Таким чином, схема демонструє послідовну трансформацію даних від початкового стану до стисненого представлення, що дозволяє наочно відобразити логіку функціонування алгоритмів обробки зображень і може бути ефективно використана для ілюстрації відповідного підрозділу у презентаційних матеріалах.

Таким чином, реалізація алгоритмів обробки зображень ґрунтується на ефективному поєднанні математичних методів і структурованої організації обчислень, що дозволяє досягти високого рівня стиснення при збереженні якості зображення. Запропонований підхід забезпечує гнучкість у налаштуванні параметрів обробки та створює передумови для подальшого вдосконалення програмного забезпечення.

3.4 Проєктування інтерфейсу програмного забезпечення для перетворення та стиснення зображень

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перетворення та стиснення зображень виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для перетворення та стиснення зображень враховані функціональні вимоги до програми:

- підтримка можливості завантаження зображень у різних форматах у програмне забезпечення;
- підтримка можливості вибору методів перетворення та стиснення зображень, таких як вейвлети Хаара та Добеши;
- можливість перегляду оригінального та обробленого зображення;
- підтримка відображення статусу процесу обробки зображення;

- підтримка можливості збереження оброблених зображень у вибраному форматі та шляхом користувача;
- підтримка ведення журналів дій користувача та обробки зображень для подальшого аналізу та відстеження помилок.

Інтерфейс програмного забезпечення для перетворення та стиснення зображень реалізовано у вигляді єдиної форми (рис. 3.4), що забезпечує цілісність взаємодії користувача із системою та спрощує процес виконання основних операцій. Організація елементів інтерфейсу побудована таким чином, щоб забезпечити інтуїтивно зрозумілий доступ до всіх функціональних можливостей програми без необхідності переходу між різними вікнами або режимами.

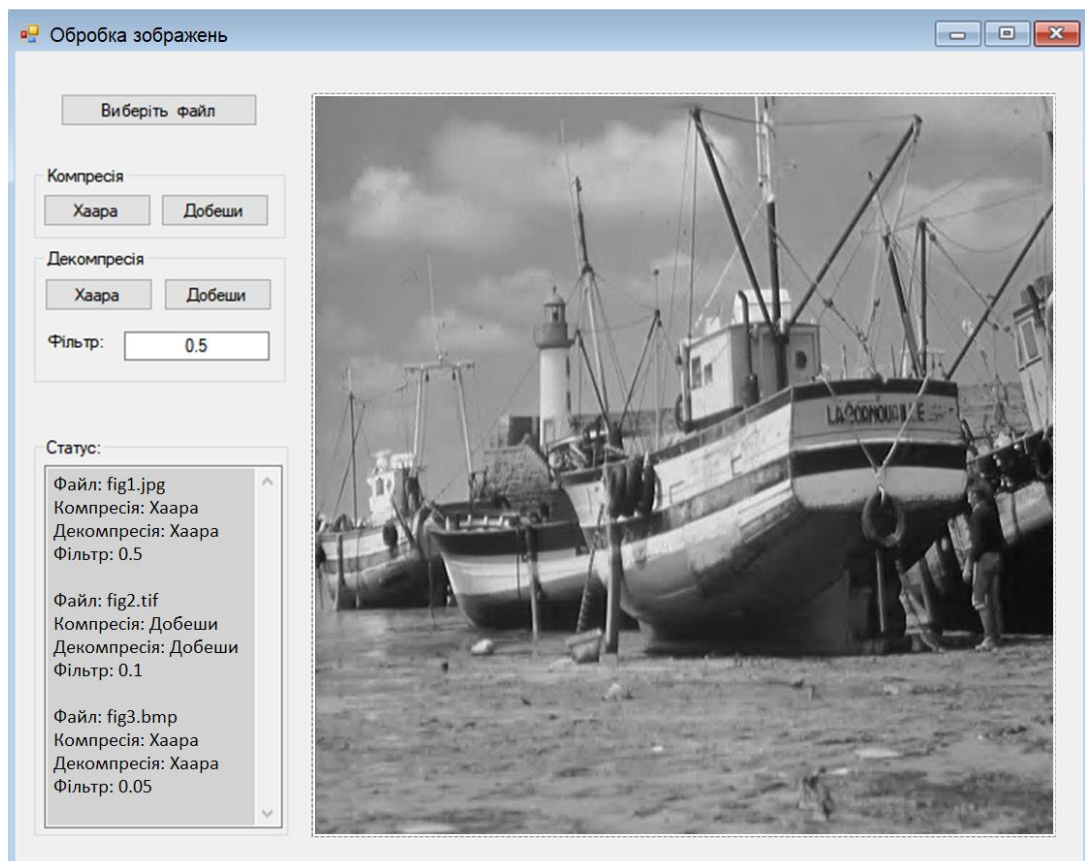


Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для перетворення та стиснення зображень

У верхній частині форми розташовано елемент керування типу кнопка, призначений для завантаження графічного файлу. Після активації відповідної кнопки відкривається стандартне діалогове вікно вибору файлу, що дозволяє користувачу обрати зображення з файлової системи. Завантажений файл одразу стає доступним для подальшої обробки, а також відображається у відповідній області інтерфейсу.

Нижче розміщено область вибору способу обробки, яка дозволяє визначити алгоритм стиснення або декомпресії. У цій частині інтерфейсу користувач має можливість обрати один із доступних методів, що реалізуються у програмі, зокрема варіанти на основі вейвлет-перетворень. Такий підхід забезпечує гнучкість у використанні програмного забезпечення та дозволяє адаптувати процес обробки до конкретних вимог.

Додатково передбачено текстове поле, у якому задається параметр, що впливає на процес обробки зображення, зокрема розмір або коефіцієнт фільтра. Введене значення використовується під час виконання алгоритмів і дозволяє змінювати рівень деталізації або ступінь стиснення. Це поле забезпечує можливість тонкого налаштування параметрів без ускладнення інтерфейсу.

Важливим елементом інтерфейсу є область відображення зображення, яка займає праву частину форми. У цій області демонструється як завантажене, так і оброблене зображення, що дозволяє користувачу безпосередньо оцінювати результати виконання алгоритмів. Візуалізація результатів у реальному часі підвищує зручність роботи та сприяє швидкому прийняттю рішень щодо подальших дій.

У нижній лівій частині форми розташовано область статусу, яка виконує функцію журналу роботи програми. У цій області відображаються повідомлення про виконані дії, поточний стан процесу обробки, а також можливі помилки або попередження. Такий підхід забезпечує прозорість функціонування системи та дозволяє користувачу контролювати всі етапи взаємодії з програмою.

Таким чином, інтерфейс програмного забезпечення характеризується простотою, логічною структурою та орієнтацією на зручність користувача. Об'єднання всіх ключових елементів на одній формі дозволяє ефективно виконувати операції завантаження, налаштування, обробки та перегляду результатів, забезпечуючи цілісний і зрозумілий процес роботи із зображеннями.

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для перетворення та стиснення зображень.

Запропоновано структуру програмного забезпечення для перетворення та стиснення зображень, яка характеризується модульністю, чітким розподілом функцій і орієнтацією на зручність використання та ефективність обробки зображень.. Визначено, що до складу ПЗ входять такі модулі: модуль користувацького інтерфейсу, модуль керування, модуль завантаження та збереження даних, модуль обробки зображень, модуль відображення та оновлення стану процесу, модуль обробки помилок і логування. Такий підхід забезпечує гнучкість у розвитку системи, спрощує її модернізацію та дозволяє досягти високого рівня якості кінцевого програмного продукту.

Описано функціонування програмного забезпечення для перетворення та стиснення зображень. Функціонування програмного забезпечення характеризується чітко визначеною послідовністю дій, що забезпечує логічну цілісність процесу обробки зображень, передбачає перевірку вхідних даних, виконання обчислювальних операцій та надання користувачу результатів у зручній формі, що робить функціонування програми зрозумілим та ефективним у практичному використанні.

Описано особливості реалізації алгоритмів обробки зображень у програмному забезпеченні, що ґрунтуються на ефективному поєднанні математичних методів і структурованої організації обчислень, що дозволяє

досягти високого рівня стиснення при збереженні якості зображення. Це забезпечує гнучкість у налаштуванні параметрів обробки та створює передумови для подальшого вдосконалення програмного забезпечення.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перетворення та стиснення зображень. Інтерфейс програми характеризується простотою, логічною структурою та орієнтацією на зручність користувача. Об'єднання всіх ключових елементів на одній формі дозволяє ефективно виконувати операції завантаження, налаштування, обробки та перегляду результатів, забезпечуючи цілісний і зрозумілий процес роботи із зображеннями.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для перетворення та стиснення зображень. Програма написана на мові C#, що обумовлюється поєднанням високого рівня абстракції, продуктивності та широких можливостей інтеграції з сучасними технологічними платформами. В якості середовища розробки для створення програмного забезпечення для перетворення та стиснення зображень обрано Visual Studio, що забезпечує високий рівень автоматизації процесів створення програм, підвищує якість коду та дозволяє ефективно реалізовувати складні алгоритми обробки графічної інформації.

Програмне забезпечення для перетворення та стиснення зображень забезпечує виконання таких функцій:

- підтримка можливості завантаження зображень у різних форматах у програмне забезпечення;
- підтримка можливості вибору методів перетворення та стиснення зображень, таких як вейвлети Хаара та Добеши;
- можливість перегляду оригінального та обробленого зображення;
- підтримка відображення статусу процесу обробки зображення;
- підтримка можливості збереження оброблених зображень у вибраному форматі та шляхом користувача;
- підтримка ведення журналів дій користувача та обробки зображень для подальшого аналізу та відстеження помилок.

Для роботи програмного забезпечення для перетворення та стиснення зображень потрібні такі технічні та програмні ресурси: комп'ютерна система, здатна забезпечити виконання операцій обробки графічних даних у прийнятний час без втрати стабільності роботи, доцільним є використання центрального процесора з тактовою частотою не нижче 2,0 ГГц, мінімально допустимий обсяг оперативної пам'яті 4 ГБ, що дозволяє уникнути затримок,

пов'язаних із використанням дискової підкачки, та забезпечує плавність виконання обчислювальних процедур.

Для зберігання вхідних і вихідних файлів необхідна наявність вільного дискового простору, обсяг якого залежить від кількості та розміру оброблюваних зображень. Мінімальні вимоги становлять 1 ГБ, однак у практичних умовах доцільно передбачити декілька гігабайтів вільного простору для забезпечення безперебійної роботи та можливості збереження результатів обробки.

З огляду на те, що програмне забезпечення реалізовано у вигляді настільного застосунку для операційної системи Microsoft Windows, необхідною умовою є наявність встановленої сучасної версії цієї операційної системи, починаючи з Windows 10 або новіших версій. Це забезпечує підтримку сучасних бібліотек і середовищ виконання, необхідних для коректної роботи програми.

Програмне середовище також передбачає наявність встановленої платформи .NET відповідної версії, яка забезпечує виконання коду, управління ресурсами та взаємодію з операційною системою.

Додатково слід враховувати наявність графічної підсистеми, що підтримує відображення інтерфейсу користувача та оброблених зображень.

Таким чином, визначені апаратні та програмні вимоги забезпечують стабільне та ефективне функціонування програмного забезпечення, дозволяючи виконувати операції перетворення та стиснення зображень із належним рівнем продуктивності та якості результатів.

Функціональні характеристики розробленого програмного забезпечення для перетворення та стиснення зображень наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для перетворення та стиснення зображень наведено у додатку Б.

4.2 Характеристики програми для перетворення та стиснення зображень

Характеристики програмного забезпечення для перетворення та стиснення зображень визначаються його здатністю ефективно виконувати обробку графічних даних, забезпечуючи поєднання якості результату, швидкодії та зручності використання. У такому програмному продукті реалізується механізм перетворення зображень, який дозволяє змінювати їх внутрішнє представлення з метою подальшої оптимізації зберігання та передачі. Особливістю є орієнтація на математичні методи обробки, що забезпечують глибокий аналіз структури зображення та виділення його ключових характеристик.

Важливою характеристикою виступає здатність до адаптивного стиснення, за якого зменшення обсягу даних досягається без істотної втрати візуальної якості. У межах програмного забезпечення реалізується обробка зображення як сукупності значень інтенсивності, що дозволяє виконувати трансформації на рівні окремих елементів. Такий підхід забезпечує гнучкість у налаштуванні параметрів обробки та дозволяє враховувати особливості конкретного зображення.

Характерною рисою програмного забезпечення для перетворення та стиснення зображень є модульність побудови, що передбачає розділення функціональності на окремі логічні компоненти. Це забезпечує можливість незалежного вдосконалення окремих частин системи та спрощує її супровід. Взаємодія між модулями організовується таким чином, щоб забезпечити послідовну передачу даних між етапами обробки, зберігаючи цілісність інформації на кожному кроці.

Значну роль відіграє ефективність використання ресурсів, оскільки обробка зображень пов'язана з виконанням великої кількості обчислювальних операцій. Водночас забезпечується можливість масштабування, що дає змогу адаптувати програму до різних умов експлуатації.

Не менш важливою характеристикою програмного забезпечення для перетворення та стиснення зображень є інтерактивність, яка проявляється у здатності системи оперативно реагувати на дії користувача та відображати результати обробки. Наявність візуального представлення зображення до та після обробки дозволяє оцінити ефективність застосованих методів і прийняти рішення щодо подальших дій.

Іншою характеристикою є надійність функціонування, що досягається за рахунок обробки помилок та контролю коректності вхідних даних. Це забезпечує стабільність роботи програмного забезпечення для перетворення та стиснення зображень та знижує ймовірність виникнення некоректних результатів.

Таким чином, програмне забезпечення для перетворення та стиснення зображень характеризується поєднанням ефективних алгоритмів обробки, оптимізованого використання ресурсів і зручності взаємодії з користувачем, що дозволяє реалізувати якісне перетворення та стиснення зображень відповідно до сучасних вимог до цифрових технологій.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для перетворення та стиснення зображень необхідно запустити відповідний виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для перетворення та стиснення зображень є графічні файли, що завантажуються користувачем із файлової системи та містять візуальну інформацію у вигляді піксельного представлення. Окрім самих зображень, до вхідних даних належать параметри обробки, що задаються користувачем, включаючи вибір методу стиснення.

Вихідними даними програмного забезпечення для перетворення та стиснення зображень є оброблені графічні файли, що отримуються внаслідок виконання алгоритмів перетворення зображень.

4.3.3 Повідомлення

Користувач програмного забезпечення для перетворення та стиснення зображень може отримати такі повідомлення: інформування про успішне завантаження зображення та готовність до обробки, повідомлення про початок і завершення процесу стиснення, а також відображення поточного стану виконання операцій.

4.4 Виконання програмного забезпечення для перетворення та стиснення зображень

Після запуску програмного забезпечення для перетворення та стиснення зображень користувачу відображається головна форма (рис. 4.1). Користувач за допомогою відповідної кнопки обирає файл із зображенням, після чого зображення відображається у правій області форми.

Після вибору зображення обирається спосіб перетворення зображення (за допомогою вейвлету Хаара або Добеши), після чого програма виконує необхідні перетворення та в режимі онлайн відображає результати (рис. 4.2). При цьому в області статусу відображається журнал роботи з програмою, де наводиться інформація про файли з зображеннями, з якими працює користувач, а також послідовність його дій над цими зображеннями.

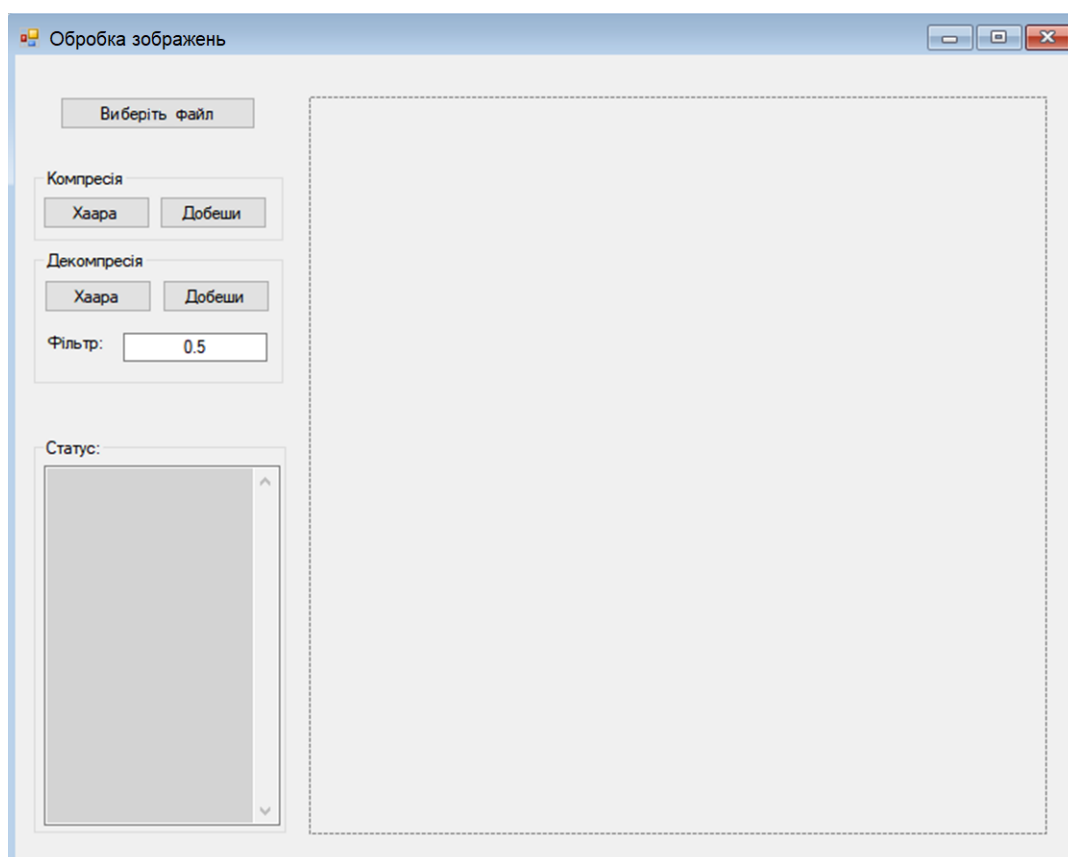


Рисунок 4.1 – Інтерфейс головної форми програмного забезпечення

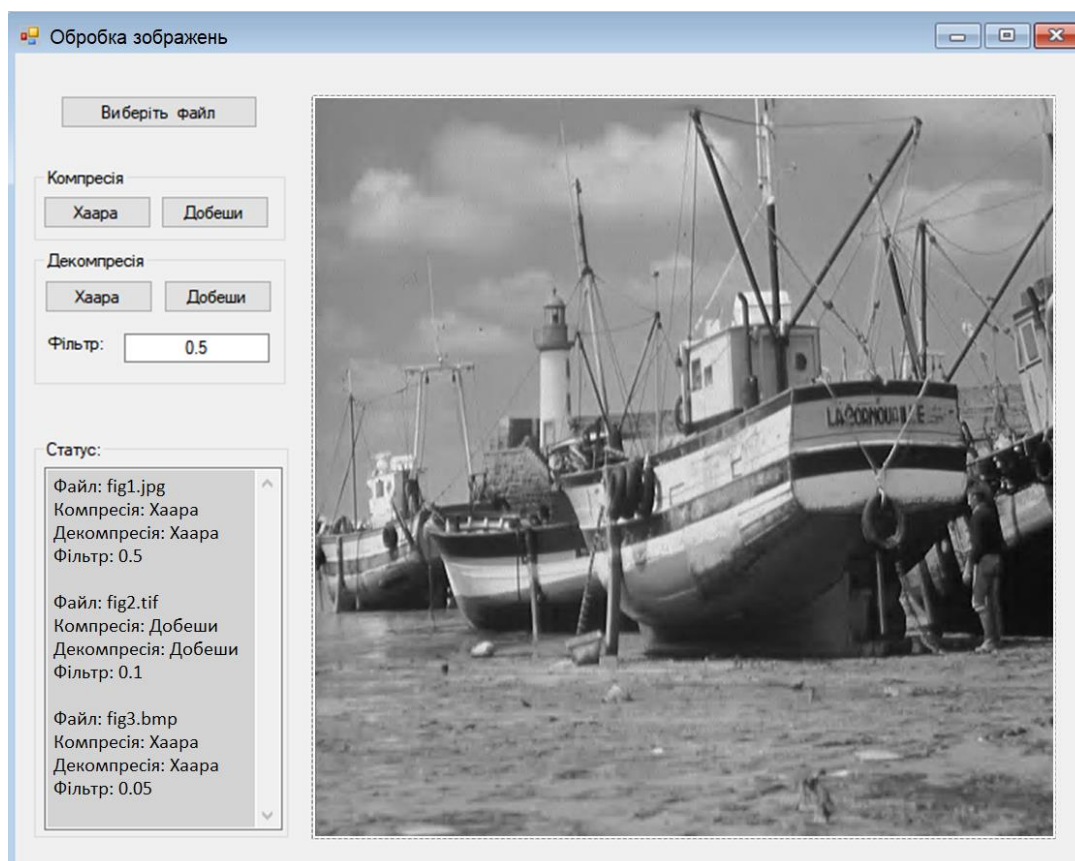


Рисунок 4.2 – Результати перетворення зображень

4.5 Тестування програмного забезпечення для перетворення та стиснення зображень

Виконано тестування програмного забезпечення для перетворення та стиснення зображень.

Виявлено, що програма для перетворення та стиснення зображень функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу перетворення та стиснення зображень.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу перетворення та стиснення зображень.

4.6 Висновки за розділом 4

Описано програмне забезпечення для перетворення та стиснення зображень. Виконано тестування розробленого програмного забезпечення для перетворення та стиснення зображень. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу перетворення та стиснення зображень.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для обробки зображень.

Визначено, що програмне забезпечення для перетворення та стиснення зображень являє собою клас програмних систем, призначених для обробки графічних файлів з метою зміни їх формату, розміру або зменшення обсягу зберігання без значної втрати якості. Такі програмні системи дозволяють оптимізувати зображення для подальшого використання у вебсайтах, мультимедійних проєктах або базах даних, підвищуючи швидкість завантаження та економлячи ресурси сховища. В їх основі лежать алгоритми обчислювальної обробки, що аналізують піксельні дані, визначають повторювані або непотрібні елементи та ефективно перетворюють інформацію у стислий формат.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для перетворення та стиснення зображень. Визначено, що недоліки існуючого програмного забезпечення для перетворення та стиснення зображень проявляються у низці аспектів, пов'язаних як з технічними обмеженнями алгоритмів, так і з особливостями їх практичної реалізації. У процесі обробки графічних даних часто спостерігається компроміс між ступенем стиснення та якістю зображення, внаслідок чого при інтенсивному зменшенні обсягу файлу виникають втрати деталей, поява артефактів і спотворення кольорів. Це особливо критично у випадках, коли зображення використовуються для професійних або наукових цілей, де точність відображення має принципове значення. Суттєвим недоліком є також залежність ефективності стиснення від типу зображення та обраного формату. Алгоритми, що добре працюють для фотографій, можуть виявитися неефективними для графіки з чіткими контурами або текстом, і навпаки. У результаті виникає необхідність додаткового налаштування

параметрів або використання різних інструментів для різних задач, що ускладнює процес автоматизації та знижує універсальність програмного забезпечення. Не менш важливим є питання сумісності та стандартизації. Існування великої кількості форматів і варіацій їх реалізації призводить до ситуацій, коли певні програми некоректно обробляють окремі файли або не підтримують новітні стандарти. Це створює додаткові труднощі при інтеграції різних програмних систем та обміні даними між ними. Тому актуальною є розробка програмного забезпечення для перетворення та стиснення зображень.

Сформульовано функціональні вимоги до програмного забезпечення для перетворення та стиснення зображень.

Для реалізації програмного забезпечення для перетворення та стиснення зображень обрано мову програмування C#, що обумовлюється поєднанням високого рівня абстракції, продуктивності та широких можливостей інтеграції з сучасними технологічними платформами. Крім того, у середовищі .NET забезпечується доступ до розвиненої екосистеми бібліотек і засобів, що спрощують реалізацію складних алгоритмів обробки графічних даних і водночас дозволяють зосередитися на логіці застосунку, а не на низькорівневих деталях керування пам'яттю чи потоками виконання.

Для створення програмного забезпечення для перетворення та стиснення зображень обрано середовище розробки Visual Studio, що забезпечує високий рівень автоматизації процесів створення програм, підвищує якість коду та дозволяє ефективно реалізовувати складні алгоритми обробки графічної інформації. Крім того, середовище Visual Studio надає зручні засоби налагодження програм, які дозволяють детально відстежувати виконання програми, аналізувати значення змінних та перевіряти коректність роботи окремих модулів.

Розроблено програмне забезпечення для перетворення та стиснення зображень.

Запропоновано структуру програмного забезпечення для перетворення та стиснення зображень, яка характеризується модульністю, чітким розподілом функцій і орієнтацією на зручність використання та ефективність обробки зображень.. Визначено, що до складу ПЗ входять такі модулі: модуль користувацького інтерфейсу, модуль керування, модуль завантаження та збереження даних, модуль обробки зображень, модуль відображення та оновлення стану процесу, модуль обробки помилок і логування. Такий підхід забезпечує гнучкість у розвитку системи, спрощує її модернізацію та дозволяє досягти високого рівня якості кінцевого програмного продукту.

Описано функціонування програмного забезпечення для перетворення та стиснення зображень. Функціонування програмного забезпечення характеризується чітко визначеною послідовністю дій, що забезпечує логічну цілісність процесу обробки зображень, передбачає перевірку вхідних даних, виконання обчислювальних операцій та надання користувачу результатів у зручній формі, що робить функціонування програми зрозумілим та ефективним у практичному використанні.

Описано особливості реалізації алгоритмів обробки зображень у програмному забезпеченні, що ґрунтуються на ефективному поєднанні математичних методів і структурованої організації обчислень, що дозволяє досягти високого рівня стиснення при збереженні якості зображення. Це забезпечує гнучкість у налаштуванні параметрів обробки та створює передумови для подальшого вдосконалення програмного забезпечення.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для перетворення та стиснення зображень. Інтерфейс програми характеризується простотою, логічною структурою та орієнтацією на зручність користувача. Об'єднання всіх ключових елементів на одній формі дозволяє ефективно виконувати операції завантаження, налаштування, обробки та перегляду результатів, забезпечуючи цілісний і зрозумілий процес роботи із зображеннями.

Виконано тестування розробленого програмного забезпечення для перетворення та стиснення зображень. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу перетворення та стиснення зображень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is image compression. URL: <https://www.cloudflare.com/learning/performance/glossary/what-is-image-compression/> (date of access: 08.05.2026).
2. Complete Image Compression Guide: Reduce File Size Without Losing Quality. URL: <https://pixcloak.com/guides/complete-image-compression-guide> (date of access: 08.05.2026).
3. Image Processing by Compression: An Overview. URL: https://hal.science/hal-00745204v1/file/Image_Processing_by_Compression_-_An_Overview_-_v2.pdf (date of access: 08.05.2026).
4. Image Compression Concepts Overview. URL: <https://www.igi-global.com/chapter/image-compression-concepts-overview/13822> (date of access: 08.05.2026).
5. What is Image Compression. URL: <https://www.educative.io/blog/what-is-image-compression> (date of access: 08.05.2026).
6. Image Compression Algorithms. URL: <https://compress.im/docs/image-compression-algorithms> (date of access: 08.05.2026).
7. What is Image Compression. URL: <https://www.geeksforgeeks.org/machine-learning/what-is-image-compression/> (date of access: 08.05.2026).
8. Best Image Compression Software Tools. URL: <https://softwarehope.com/15-best-image-compression-software-tools-compared/> (date of access: 08.05.2026).
9. The Top Free Image Compressors for Optimizing Your Photos. URL: <https://enviragallery.com/best-free-image-optimization-tools-for-image-compression/> (date of access: 08.05.2026).
10. Best Image Optimizers. URL: <https://fixthephoto.com/best-image-optimizer.html> (date of access: 08.05.2026).

11. The Most Recommended Free Image Compression Software. URL: <https://www.geekersoft.com/resources-multimedia-solutions/the-10-most-recommended-free-image-compression-softwares.html> (date of access: 08.05.2026).
12. Best Free Image Converters. URL: <https://www.movavi.com/learning-portal/best-free-image-converter.html> (date of access: 08.05.2026).
13. Best Image Converters. URL: <https://sourceforge.net/software/image-converters/> (date of access: 08.05.2026).
14. EWWW Image Optimization. URL: <https://ewww.io/features/> (date of access: 08.05.2026).
15. ImageRecycle. URL: <https://www.imagerecycle.com/> (date of access: 08.05.2026).
16. Zamzar. URL: <https://www.zamzar.com/> (date of access: 08.05.2026).
17. C# Tutorial. URL: <https://www.csharptutorial.net/> (date of access: 08.05.2026).
18. C# Tutorials. URL: <https://www.c-sharpcorner.com/csharp-tutorials> (date of access: 08.05.2026).
19. Getting started with Visual Studio IDE. URL: <https://visualstudio.microsoft.com/vs/getting-started/> (date of access: 08.05.2026).
20. Visual Studio vs Visual Studio Code - What to Choose. URL: <https://www.geeksforgeeks.org/blogs/visual-studio-vs-visual-studio-code/> (date of access: 08.05.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для перетворення та стиснення зображень.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для перетворення та стиснення зображень», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

А.2 Призначення розробки

Програмний продукт призначений для оброблення зображень.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для перетворення та стиснення зображень забезпечує виконання таких функцій:

- підтримка можливості завантаження зображень у різних форматах у програмне забезпечення;
- підтримка можливості вибору методів перетворення та стиснення зображень, таких як вейвлети Хаара та Добеши;
- можливість перегляду оригінального та обробленого зображення;
- підтримка відображення статусу процесу обробки зображення;
- підтримка можливості збереження оброблених зображень у вибраному форматі та шляхом користувача;

– підтримка ведення журналів дій користувача та обробки зображень для подальшого аналізу та відстеження помилок.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для перетворення та стиснення зображень повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

A.3.3 Вимоги до надійності

Програмне забезпечення для перетворення та стиснення зображень повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для перетворення та стиснення зображень необхідна наявність персонального комп'ютера.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для перетворення та стиснення зображень потрібні такі технічні та програмні ресурси: комп'ютерна система, здатна забезпечити виконання операцій обробки графічних даних у прийнятний час без втрати стабільності роботи, доцільним є використання центрального процесора з тактовою частотою не нижче 2,0 ГГц, мінімально допустимий обсяг оперативної пам'яті 4 ГБ, що дозволяє уникнути затримок,

пов'язаних із використанням дискової підкачки, та забезпечує плавність виконання обчислювальних процедур.

Для зберігання вхідних і вихідних файлів необхідна наявність вільного дискового простору, обсяг якого залежить від кількості та розміру оброблюваних зображень. Мінімальні вимоги становлять 1 ГБ, однак у практичних умовах доцільно передбачити декілька гігабайтів вільного простору для забезпечення безперебійної роботи та можливості збереження результатів обробки.

З точки зору програмного забезпечення, необхідною умовою є наявність встановленої операційної системи Windows 10 або новіших версій.

А.3.6 Вимоги до маркування і пакування

Програма для перетворення та стиснення зображень може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для перетворення та стиснення зображень».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для перетворення та стиснення зображень є графічні файли, що завантажуються користувачем із файлової системи та містять візуальну інформацію у вигляді піксельного представлення. Окрім самих зображень, до вхідних даних належать параметри обробки, що задаються користувачем, включаючи вибір методу стиснення.

Вихідними даними програмного забезпечення для перетворення та стиснення зображень є оброблені графічні файли, що отримуються внаслідок виконання алгоритмів перетворення зображень.

ДОДАТОК Б
Фрагмент тексту програми

```

ImgPrssngWinForms
{
    public class MnFrm : Form
    {
        private Bitmap srcimg;
        private Bitmap dstimg;
        private PictureBox pbxImg;
        private ProgressBar prgSts;

        public MnFrm()
        {
            this.Text = "Img Prssng App";
            this.ClientSize = new Size(800, 600);

            pbxImg = new PictureBox();
            pbxImg.Dock = DockStyle.Fill;
            this.Controls.Add(pbxImg);

            prgSts = new ProgressBar();
            prgSts.Dock = DockStyle.Bottom;
            this.Controls.Add(prgSts);
        }

        public void lddfl(string pthfl)
        {
            if (!File.Exists(pthfl))
                throw new Exception("Fl nt fnd");

            using (var strm = new FileStream(pthfl,
FileMode.Open, FileAccess.Read))
            {
                srcimg = new Bitmap(strm);
            }

            if (srcimg.Width == 0 || srcimg.Height == 0)
                throw new Exception("Nt vld img");

            pbxImg.Image = srcimg;
        }

        public void prcssimg(string mthd)
        {
            if (srcimg == null)
                throw new Exception("N src");

            double[, ] mtrx = ImgCnvrtr.cnvrt(srcimg);

```

```

double[,] trns;

if (mthd == "hr")
    trns = HrrWvlt.prcss(mtrx);
else
    trns = DbchWvlt.prcss(mtrx);

double[,] cmprs = CmprsMdl.cmprss(trns);

double[,] rcns = RcnstrMdl.rcnstrct(cmprs, mthd);

dstimg = ImgCnvrtr.rstr(rcns);

pbxImg.Image = dstimg;
}

public void svrslt(string pthfl)
{
    if (dstimg == null)
        throw new Exception("N rslt");

    dstimg.Save(pthfl);
}
}

public static class ImgCnvrtr
{
    public static double[,] cnvrt(Bitmap bmp)
    {
        int wdh = bmp.Width;
        int hgt = bmp.Height;

        double[,] rslt = new double[hgt, wdh];

        for (int r = 0; r < hgt; r++)
        {
            for (int c = 0; c < wdh; c++)
            {
                Color clr = bmp.GetPixel(c, r);
                double gry = (clr.R + clr.G + clr.B) / 3.0;
                rslt[r, c] = gry;
            }
        }

        return rslt;
    }

    public static Bitmap rstr(double[,] mtrx)

```

```

    {
        int hght = mtrx.GetLength(0);
        int wtdh = mtrx.GetLength(1);

        Bitmap bmp = new Bitmap(wtdh, hght);

        for (int r = 0; r < hght; r++)
        {
            for (int c = 0; c < wtdh; c++)
            {
                int v = (int)Math.Max(0, Math.Min(255,
mtrx[r, c]));
                bmp.SetPixel(c, r, Color.FromArgb(v, v, v));
            }
        }

        return bmp;
    }
}

public static class HrrWvlt
{
    public static double[,] prcss(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

        double[,] tmp = new double[n, m];

        for (int r = 0; r < n; r++)
        {
            for (int c = 0; c < m; c += 2)
            {
                double a = src[r, c];
                double b = src[r, c + 1];

                tmp[r, c / 2] = (a + b) / 2;
                tmp[r, (c / 2) + (m / 2)] = (a - b) / 2;
            }
        }

        double[,] rslt = new double[n, m];

        for (int c = 0; c < m; c++)
        {
            for (int r = 0; r < n; r += 2)
            {
                double a = tmp[r, c];

```

```

        double b = tmp[r + 1, c];

        rslt[r / 2, c] = (a + b) / 2;
        rslt[(r / 2) + (n / 2), c] = (a - b) / 2;
    }
}

return rslt;
}
}

public static class DbchWvlt
{
    public static double[,] prcss(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

        double[] fltr = new double[] { 0.4829, 0.8365,
0.2241, -0.1294 };

        double[,] rslt = new double[n, m];

        for (int r = 0; r < n; r++)
        {
            for (int c = 0; c < m; c++)
            {
                double sm = 0;
                for (int k = 0; k < fltr.Length; k++)
                {
                    int idx = (c + k) % m;
                    sm += src[r, idx] * fltr[k];
                }
                rslt[r, c] = sm;
            }
        }

        return rslt;
    }
}

public static class CmprsMdl
{
    public static double[,] cmprss(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

```

```

double[,] rslt = new double[n, m];

double thr = clctthr(src);

for (int r = 0; r < n; r++)
{
    for (int c = 0; c < m; c++)
    {
        double v = src[r, c];
        rslt[r, c] = Math.Abs(v) < thr ? 0 : v;
    }
}

return rslt;
}

static double clctthr(double[,] src)
{
    double sm = 0;
    int cnt = 0;

    foreach (var v in src)
    {
        sm += Math.Abs(v);
        cnt++;
    }

    return sm / cnt;
}

}

public static class RcnstrMdl
{
    public static double[,] rcnstrct(double[,] src, string
mthd)
    {
        if (mthd == "hr")
            return rcnstrhr(src);
        else
            return rcnstrdb(src);
    }

    static double[,] rcnstrhr(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

        double[,] rslt = new double[n, m];

```

```

        for (int r = 0; r < n / 2; r++)
        {
            for (int c = 0; c < m / 2; c++)
            {
                double a = src[r, c];
                double d = src[r, c + m / 2];

                rslt[r * 2, c * 2] = a + d;
                rslt[r * 2, c * 2 + 1] = a - d;
            }
        }

        return rslt;
    }

    static double[,] rcnstrdb(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

        double[,] rslt = new double[n, m];

        for (int r = 0; r < n; r++)
        {
            for (int c = 0; c < m; c++)
            {
                rslt[r, c] = src[r, c];
            }
        }

        return rslt;
    }
}

static class Prgrm
{
    [STAThread]
    static void Mn()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);

        MnFrm frm = new MnFrm();
        Application.Run(frm);
    }
}

```

```

public static class DbchWvltExt
{
    public static double[,] advprcss(double[,] src, int lvl)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);

        double[,] rslt = (double[,])src.Clone();

        double[] fltrLow = new double[] { 0.4829, 0.8365,
0.2241, -0.1294 };
        double[] fltrHigh = new double[] { -0.1294, -0.2241,
0.8365, -0.4829 };

        for (int l = 0; l < lvl; l++)
        {
            for (int r = 0; r < n; r++)
            {
                double[] tmp = new double[m];
                for (int c = 0; c < m; c++)
                {
                    double sm = 0;
                    for (int k = 0; k < fltrLow.Length; k++)
                    {
                        int idx = (c + k) % m;
                        sm += rslt[r, idx] * fltrLow[k];
                    }
                    tmp[c] = sm;
                }

                for (int c = 0; c < m; c++)
                {
                    double smh = 0;
                    for (int k = 0; k < fltrHigh.Length; k++)
                    {
                        int idx = (c + k) % m;
                        smh += tmp[idx] * fltrHigh[k];
                    }
                    rslt[r, c] = smh;
                }
            }
        }

        return rslt;
    }
}

public static class SttsMdl

```

```

    {
        public static void UpdPrgrs(ProgressBar prg, double
prcnt)
        {
            int val = (int)Math.Max(0, Math.Min(100, prcnt));
            prg.Value = val;
        }

        public static void ShwMsg(Form frm, string txt)
        {
            MessageBox.Show(frm, txt, "Статус процесу",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

    public static class CmprsMdlExt
    {
        public static double[,] thrshldcmprss(double[,] src,
double pct)
        {
            int n = src.GetLength(0);
            int m = src.GetLength(1);

            double thr = clctthr(src) * pct;
            double[,] rslt = new double[n, m];

            for (int r = 0; r < n; r++)
            {
                for (int c = 0; c < m; c++)
                {
                    rslt[r, c] = Math.Abs(src[r, c]) < thr ? 0 :
src[r, c];
                }
            }

            return rslt;
        }

        private static double clctthr(double[,] src)
        {
            double sm = 0;
            int cnt = 0;

            foreach (var v in src)
            {
                sm += Math.Abs(v);
                cnt++;
            }
        }
    }

```

```

        return sm / cnt;
    }
}

public static class ImgUtils
{
    public static double CalcAvgBrightness(Bitmap bmp)
    {
        double sm = 0;
        int cnt = 0;

        for (int r = 0; r < bmp.Height; r++)
        {
            for (int c = 0; c < bmp.Width; c++)
            {
                Color clr = bmp.GetPixel(c, r);
                sm += (clr.R + clr.G + clr.B) / 3.0;
                cnt++;
            }
        }

        return sm / cnt;
    }

    public static Bitmap ApplyContrast(Bitmap bmp, double
factor)
    {
        Bitmap rslt = new Bitmap(bmp.Width, bmp.Height);

        for (int r = 0; r < bmp.Height; r++)
        {
            for (int c = 0; c < bmp.Width; c++)
            {
                Color clr = bmp.GetPixel(c, r);
                int R = Math.Max(0, Math.Min(255,
(int)((clr.R - 128) * factor + 128)));
                int G = Math.Max(0, Math.Min(255,
(int)((clr.G - 128) * factor + 128)));
                int B = Math.Max(0, Math.Min(255,
(int)((clr.B - 128) * factor + 128)));
                rslt.SetPixel(c, r, Color.FromArgb(R, G, B));
            }
        }

        return rslt;
    }
}

```

```

public static class HrrWvltExt
{
    public static double[,] cmpxprcss(double[,] src)
    {
        int n = src.GetLength(0);
        int m = src.GetLength(1);
        double[,] tmp = new double[n, m];

        for (int r = 0; r < n; r++)
        {
            for (int c = 0; c < m; c += 4)
            {
                double s0 = src[r, c];
                double s1 = src[r, c + 1];
                double s2 = src[r, c + 2];
                double s3 = src[r, c + 3];

                tmp[r, c / 4] = (s0 + s1 + s2 + s3) / 4;
                tmp[r, (c / 4) + (m / 4)] = (s0 - s1 + s2 -
s3) / 4;
                tmp[r, (c / 4) + (m / 2)] = (s0 + s1 - s2 -
s3) / 4;
                tmp[r, (c / 4) + (3 * m / 4)] = (s0 - s1 - s2
+ s3) / 4;
            }
        }

        return tmp;
    }
}

public static class RsltMdl
{
    public static void SavePreview(Bitmap bmp, string pth)
    {
        bmp.Save(pth);
        Lggng.wrtln("Preview saved to " + pth);
    }

    public static void CompareBitmaps(Bitmap bmp1, Bitmap
bmp2, out double mse)
    {
        int w = bmp1.Width;
        int h = bmp1.Height;
        double sum = 0;

        for (int r = 0; r < h; r++)

```

```

        {
            for (int c = 0; c < w; c++)
            {
                Color clr1 = bmp1.GetPixel(c, r);
                Color clr2 = bmp2.GetPixel(c, r);
                sum += Math.Pow(clr1.R - clr2.R, 2) +
Math.Pow(clr1.G - clr2.G, 2) + Math.Pow(clr1.B - clr2.B, 2);
            }

            mse = sum / (w * h * 3);
        }
    }
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Програмне забезпечення для перетворення та стиснення зображень

Виконав: Данило ПОПОВ

ст. групи КНТ-113сп

Керівник: Олена ШИТИКОВА

к.т.н., доцент

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

- Об'єкт дослідження – процес розробки програмного забезпечення для обробки зображень.
- Предмет дослідження – програмні засоби для перетворення зображень.
- Метою роботи є розробка програмного забезпечення для перетворення та стиснення зображень, що забезпечує скорочення витрат дискового простору та мережеских ресурсів при обробці графічних даних.

2

Рисунок В.2 – Слайд 2

ЗАВДАННЯ РОБОТИ

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для перетворення та стиснення зображень;
- здійснити проєктування програмного забезпечення для перетворення та стиснення зображень;
- створити програмне забезпечення для перетворення та стиснення зображень;
- виконати тестування розробленого програмного забезпечення для перетворення та стиснення зображень.

3

Рисунок В.3 – Слайд 3

ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ

Критерій порівняння	EWWW Image Optimization	ImageRecycle	Zamzar
підтримка локальної обробки зображень	+	+–	–
робота без встановлення	+–	+	+
інтеграція з системами управління вмістом	+	+	+–
можливість API-інтеграції	+–	+–	+
контроль якості стиснення	+	+	+–
простота інтерфейсу	+–	+	+

4

Рисунок В.4 – Слайд 4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

При розробці програмного забезпечення для перетворення та стиснення зображень необхідно забезпечити такі функціональні вимоги:

- підтримка можливості завантаження зображень у різних форматах у програмне забезпечення;
- підтримка можливості вибору методів перетворення та стиснення зображень, таких як вейвлети Хаара та Добеши;
- можливість перегляду оригінального та обробленого зображення;
- підтримка відображення статусу процесу обробки зображення;
- підтримка можливості збереження оброблених зображень у вибраному форматі та шляхом користувача;
- підтримка ведення журналів дій користувача та обробки зображень для подальшого аналізу та відстеження помилок.

5

Рисунок В.5 – Слайд 5

ПОРІВНЯННЯ МОВ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	C#	C++	Java
Продуктивність обробки зображень	+	+	+-
Швидкість розробки	+	-	+-
Зручність роботи з пам'яттю	+	-	+
Якість інструментів розробки	+	+-	+
Баланс продуктивності та простоти	+	-	+-

6

Рисунок В.6 – Слайд 6

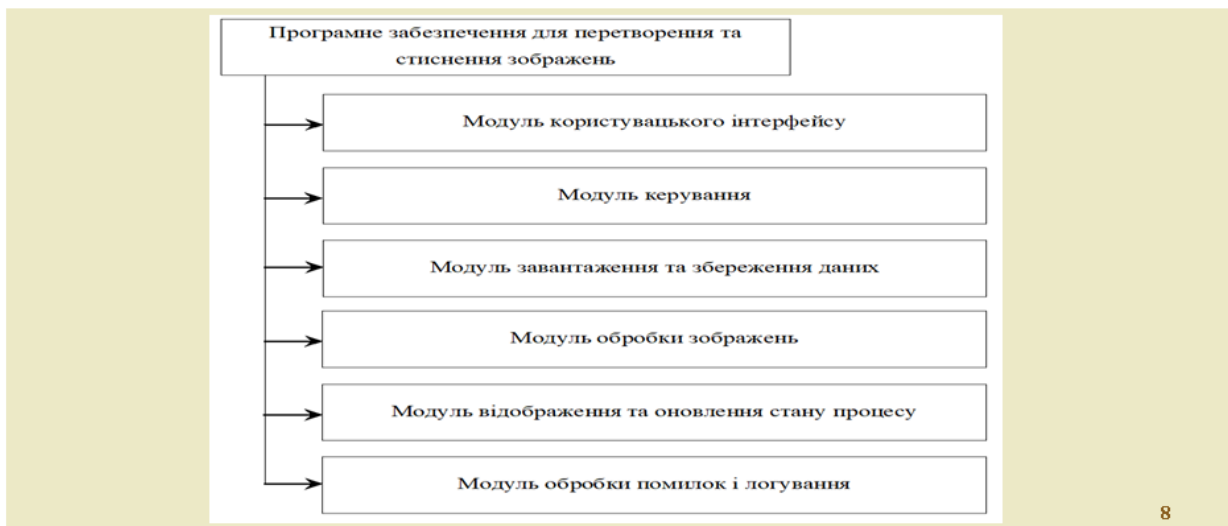
ПОРІВНЯННЯ СЕРЕДОВИЩ РОЗРОБКИ

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	JetBrains Rider	MonoDevelop
Інструменти налагодження	+	+	+—
Засоби профілювання продуктивності	+	+—	—
Інтеграція з графічними бібліотеками	+	+	+—
Швидкість роботи середовища	+	+—	+—
Зручність інтерфейсу	+	+—	+—

7

Рисунок В.7 – Слайд 7

СТРУКТУРА ПРОГРАМИ



8

Рисунок В.8 – Слайд 8

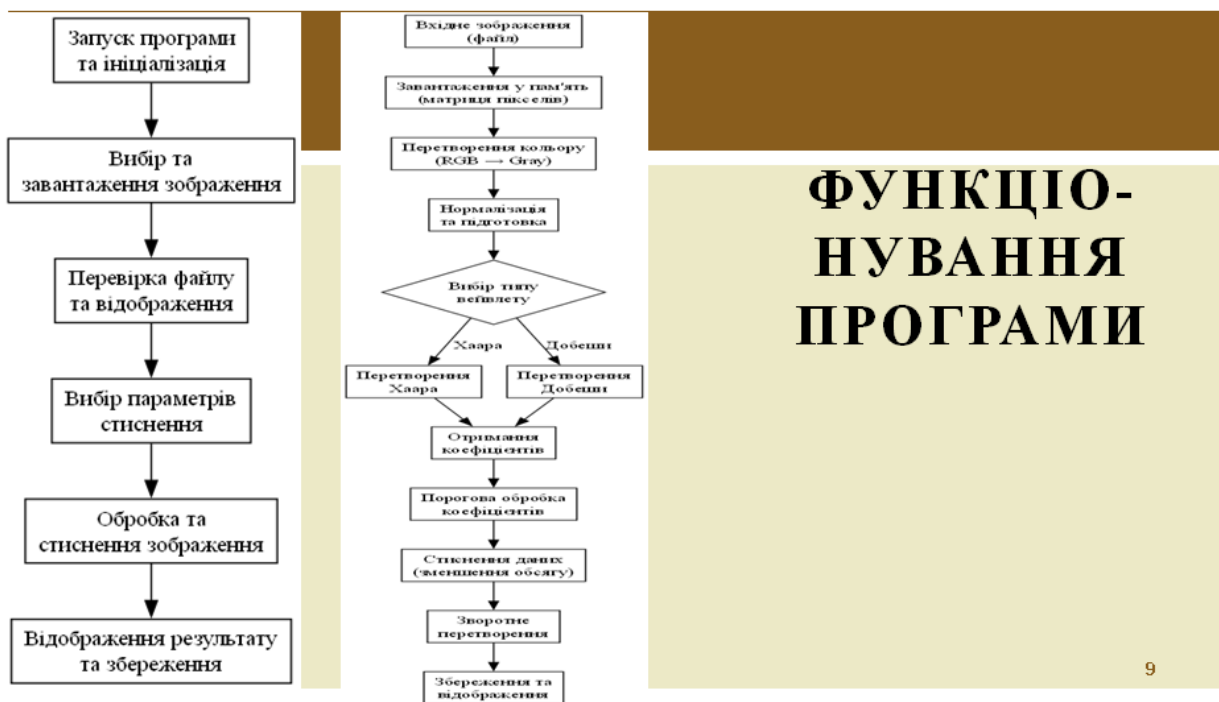


Рисунок В.9 – Слайд 9

ІНТЕРФЕЙС ЗАСТОСУНКУ

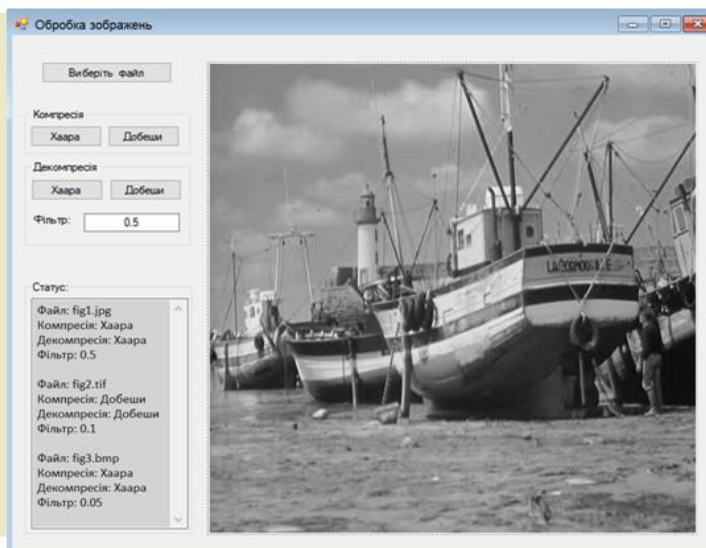


Рисунок В.10 – Слайд 10

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було розроблено програмне забезпечення для перетворення та стиснення зображень. Для цього було виконано ряд завдань:

- аналіз предметної області та програмних засобів для перетворення та стиснення зображень;
- проєктування програмного забезпечення для перетворення та стиснення зображень;
- конструювання програмного забезпечення для перетворення та стиснення зображень;
- тестування розробленого програмного забезпечення для перетворення та стиснення зображень.

Результати тестування програмного забезпечення показали, що розроблена програма відповідає технічному завданню.

11

Рисунок В.11 – Слайд 11