

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ЗАСТОСУНКУ ДЛЯ**  
**ОБРОБКИ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ**  
(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-19-2

Яковенко В.В.  
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник доц. Шафроненко А.Ю.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Кобилін О.А.  
(прізвище, ініціали)

2023 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Яковенку Віктору Володимировичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка застосунку для обробки фотореалістичних зображень

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 27 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом Emgu CV, мова програмування C#, середовище розробки Visual Studio 2022.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз основних методів обробки зображень.

2. Алгоритми обробки фотореалістичних зображень.

3. Реалізація застосунку для обробки фотореалістичних зображень.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми обробки фотореалістичних зображень, постановка задачі, сучасні методи обробки зображень, вибір методу реалізації, тестові зображення.

---



---



---



---



---



---



---



---

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-17.04.23	
3	Аналіз літератури з досліджуваної проблеми	18.04.23-20.04.23	
4	Аналіз технічних і програмних засобів	21.04.23-30.04.23	
5	Розробка методу	01.05.23-14.05.23	
6	Програмна реалізація	15.05.23-25.05.23	
7	Оформлення пояснювальної записки	26.05.23-30.05.23	
8	Перевірка на плагіат	01.06.23	
9	Рецензування	02.06.23	
10	Підготовка презентації та доповіді	03.06.23-04.06.23	
11	Занесення роботи в електронний архів	05.06.23	
12	Попередній захист кваліфікаційної роботи	05.06.23	

Дата видачі завдання 10 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Шафроненко А.Ю.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 67 с., 36 рис., 30 джерел.

ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ, ОБРОБКА ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ, ГЕОМЕТРИЧНІ ПЕРЕТВОРЕННЯ, .NET 6.0, КОНТРАСТ, ЯСКРАВІСТЬ, EMGU CV.

Об'єктом роботи є реалізація застосунку для обробки фотореалістичних зображень.

Метою роботи є розробка застосунку зі зручним і інтуїтивно зрозумілим інтерфейсом, що базується на використанні сучасних методів і алгоритмів обробки фотореалістичних зображень, який забезпечить високу якість обробки зображень.

Використано методи фільтрації зображень, налаштування контрастності, яскравості і геометричних перетворень. Проведено порівняльний аналіз результатів обробки зображень, отриманих з використанням розроблених методів, з результатами, отриманими за допомогою існуючих алгоритмів та програм.

У результаті роботи здійснена програмна реалізація застосунку для обробки фотореалістичних зображень.

IMAGE FILTERING, PHOTOREALISTIC IMAGE PROCESSING, GEOMETRIC TRANSFORMATIONS, .NET 6.0, CONTRAST, BRIGHTNESS, EMGU CV.

The object of the work is the implementation of an application for processing photorealistic images.

The purpose of the work is to develop methods and algorithms that will allow efficient and accurate processing of these images using modern approaches and technologies.

The purpose of the work is to develop an application with a convenient and intuitive interface, based on the use of modern methods and algorithms of photorealistic image processing, which will ensure high quality of image processing.

As a result of the work, the software implementation of the application for processing photorealistic images was carried out.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Аналіз основних методів обробки зображень.....	10
1.1 Поняття цифрового зображення.....	10
1.1.1 Види зображень.....	11
1.1.2 Основні поняття при обробці зображень.....	11
1.2 Геометричні перетворення.....	13
1.3 Фільтрація зображень.....	16
1.4 Сучасні застосунки для обробки фото.....	18
1.4.1 Adobe Photoshop .....	18
1.4.2 GIMP.....	19
1.4.3 Google Photos .....	19
1.5 Постановка задачі .....	20
2 Алгоритми обробки фотореалістичних зображень .....	21
2.1 Видалення шумів .....	21
2.1.1 Фільтр Гауса .....	21
2.1.2 Медіанний фільтр.....	23
2.1.3 Фільтрація за середнім значенням .....	25
2.1.4 Комбінація різних методів обробки .....	26
2.2 Колірна корекція .....	27
2.2.1 Алгоритм «сірого світу».....	28
2.2.2 Автоматичний баланс білого .....	30
2.3 Обрізка та зміна розміру .....	32
2.3.1 Обрізка .....	33
2.3.2 Масштабування зображення .....	34
3 Реалізація застосунку для обробки фотореалістичних зображень .....	38
3.1 Обґрунтування вибору середовища програмної реалізації .....	38
3.2 Програмна реалізація.....	42

	6
3.3 Інструкція користувача .....	44
3.4 Тестування розробленої моделі.....	48
3.4.1 Причини проведення тестування.....	48
3.4.2 Основні етапи тестування .....	49
3.4.3 Тестування застосунку .....	50
Висновки .....	64
Перелік джерел посилання .....	65

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RGB – Red, Green, Blue (червоний, зелений, синій)

CMYK – Cyan, Magenta, Yellow and Black (ціан, маджента, жовтий і чорний)

HSV – Hue, Saturation, Lightness (кольоровий тон, насиченість і освітлення)

Pixel – Picture Element (елемент зображення)

DPI – Dots Per Inch (точок на дюйм)

PPI – Pixels Per Inch (пікселів на дюйм)

IDE – Integrated Development Environment (інтегроване середовище розробки)

GDI – Graphics Device Interface (інтерфейс графічного пристрою)

API – Application Programming Interface (прикладний програмний інтерфейс)

КТ – ключові точки

FAST – Features From Accelerated Segment Test (ознаки з прискореної сегментної перевірки)

BRIEF – Binary Robust Independent Elementary Features (бінарні надійні незалежні елементарні функції)

ORB – Oriented FAST And Rotated BRIEF (орієнтовані FAST і обернені BRIEF)

## ВСТУП

Комп'ютерна графіка – це галузь знань, що вивчає та розробляє методи та засоби синтезу, збереження й перетворення цифрових зображень за допомогою електронно-обчислювальних машин та інших технічних пристроїв.

В комп'ютерній графіці фотореалізм забезпечує рівень реалістичності, який наближає зображення до фотографії за допомогою різних технологій, наприклад, рендеринг.

В сучасному цифровому світі фотографії є одним із основних засобів передачі інформації, привернення уваги та створення емоційного зв'язку між людьми та навколишнім світом. Однак, не завжди фотографії виходять ідеальними, а іноді потребують додаткової обробки, щоб досягти максимально реалістичного візуального ефекту. Саме через це обробка фотореалістичних зображень є надзвичайно актуальною та важливою задачею.

Цифрові зображення є важливою складовою багатьох галузей в сучасному суспільстві, таких як наука, медицина, дизайн, ЗМІ, соціальні мережі, кінематограф, відеоігри тощо.

У сфері комп'ютерного зору використовуються різні методи та алгоритми для аналізу та обробки зображень. Основним завданням комп'ютерного зору є розпізнавання візуальних об'єктів на аналізованому зображенні [1].

Багато людей використовують обробку зображень повсякденно через те, що вона дозволяє створити привабливі зображення високої якості за допомогою зміни форми та розміру, зміни кольорів та текстур. Фільтрація зображень дозволяє отримувати більш точну та детальну інформацію зі зображень, що може бути корисним у різних дослідженнях та проєктах.

Для створення привабливих та високоякісних зображень необхідні спеціальні технології та програмне забезпечення, які дозволяють покращувати якість та реалізовувати різні ідеї та концепції.

На виході обробки зображення можна отримати відредаговане зображення або вектор даних, що описує зображення. Ця інформація може використовуватись для різних цілей, таких як розпізнавання облич, виявлення ракових клітин на медичних зображеннях, розпізнавання автомобільних номерних знаків на фото-відео зйомках, безпілотних літальних апаратів, мобільних пристроїв, роботизованих та супутникових систем тощо [2].

Фотореалістичне зображення має відтворювати усі нюанси кольорів, тіні, освітлення, композицію, кут нахилу камери тощо. В результаті зображення має виглядати настільки реалістичним, щоб створювалося враження того, що це зображення є реальною фотографією.

Обробка фотореалістичних зображень покращує якість, прибирає дефекти і робить зображення більш привабливими для людського ока. Обробка фото може включати в себе такі процеси, як ретушування, збільшення різкості, чіткості, усунення шумів, він'єтування, налаштування експозиції, контрасту, насиченості, теплоти тощо.

Вже існує достатньо багато редакторів фотографій, які змінюють та покращують зображення. Проте, якість цих програм може значно відрізнятись, тому до вибору редактора потрібно підходити з особливою увагою. Редактори фотографій дають можливість відредагувати фотографії, щоб вони виглядали більш професійно, привабливіше та ефектніше, але більшість з них є або занадто дорогими, що не дозволяє звичайним людям, які не займаються професійною обробкою фото, редагувати свої фото. В деяких редакторах інтерфейс програми є досить складним для розуміння через те, що в програмі занадто багато можливостей.

В даній роботі буде розроблено програмний застосунок, заснований на методах та алгоритмах, який використовуватиметься для обробки та покращення фотореалістичних зображень.

# 1 АНАЛІЗ ОСНОВНИХ МЕТОДІВ ОБРОБКИ ЗОБРАЖЕНЬ

## 1.1 Поняття цифрового зображення

Зображення – інформація, орієнтована на зорове сприйняття. Зображення можна класифікувати за двома видами: аналогове і цифрове. Аналогове зображення використовує безперервні аналогові сигнали для передачі інформації. Цифрове зображення є матрицею точок, які називаються пікселями (pixels), які зберігають інформацію про колір, найчастіше, у трьох кольорових моделях RGB, CMYK або HSV. Зображення в комп'ютерній графіці мають два загальні класи: растрові та векторні.

Щоб краще зрозуміти властивості зображення, яке використовується для обробки, можна розглядати зображення як функцію  $f$ . Кожен піксель також має своє значення. Для зображення у градаціях сірого кожен піксель буде мати інтенсивність від 0 до 255, де 0 буде чорним, а 255 – білим. Таким чином, зображення  $f(x,y)$  можна розглядати як функцію, яка зіставляє кожній парі координат  $(x,y)$  значення інтенсивності пікселя. Значення інтенсивності зазвичай є цілими числами від 0 до 255 у монохромному зображенні та трійками чисел від 0 до 255 у кольоровому зображенні, де кожне число означає інтенсивність червоного, зеленого та синього кольорів відповідно.

Тобто зображення можна представити у вигляді функції двох змінних  $f(x,y)$ , визначену у деякій області  $S$  площини  $(x,y)$ , що має певну множину своїх значень. Наприклад, звичайну чорно-білу фотографію можна подати, як [3]:

$$f(x,y) \geq 0, 0 \leq x \leq a, 0 \leq y \leq b,$$

де  $f(x,y)$  – оптична щільність (або інтенсивність) каналу в точці  $(x, y)$ ;

$a$  – ширина кадру;

$b$  – висота кадру.

### 1.1.1 Види зображень

Растрові зображення – зображення, які складаються з пікселів, кожен із яких має свій колір і є незалежним від інших пікселів у зображенні. Вони зазвичай використовуються для фотографій та реалістичних зображень, оскільки вони можуть відтворювати безліч кольорів та відтінків. Недоліком растрових зображень є те, що вони можуть втратити якість зі збільшенням розміру, оскільки пікселі стають більш помітними. Більшість застосунків для обробки фотографій працюють саме з растровими зображеннями.

Растрове зображення представляється у вигляді великої кількості точок – чим їх більше, тим візуально якісніше зображення та більше розмір файлу. Тобто, одна й та сама картинка може бути представлена з кращою або гіршою якістю відповідно до кількості точок на одиницю довжини – роздільною здатністю (зазвичай, DPI або PPI).

Векторні зображення – це зображення, які складаються з геометричних фігур, таких як лінії, криві та багатокутники. Вони використовують математичні формули для визначення розміру, форми та кольору кожної фігури. Векторні зображення використовуються для створення логотипів, ілюстрацій та графіки, оскільки вони зберігають свою чіткість та якість при будь-якому масштабуванні та зміні розміру, на відміну від растрового зображення.

### 1.1.2 Основні поняття при обробці зображень

Як в же і було сказано раніше, основним елементом зображення є піксель. Піксель у растровому зображенні можна вказати, сказавши, який стовпець і який рядок його містять. З точки зору координат, піксель можна ідентифікувати парою цілих чисел, що дають номер стовпця та номер рядка.

Стовпці нумеруються зліва направо, починаючи з нуля. Рядки нумеруються зазвичай зверху вниз.

Роздільна здатність фото визначає кількість пікселів, які містяться в зображенні. Вона визначає рівень деталізації та якість зображення. Роздільна здатність вимірюється в РРІ і вказує, скільки точок розміщується на одному дюймі зображення. Чим вища роздільна здатність, тим більше деталей та інформації буде на зображенні. Необхідно врахувати, що підвищення роздільної здатності фото може призвести до збільшення розміру файлу зображення, що може бути проблематичним під час передачі або зберігання. Крім того, збільшення роздільної здатності фото не завжди покращує якість зображення, якщо при цьому порушуються оригінальні пропорції та співвідношення сторін фотографії.

Іншим важливим визначенням є кольорова модель зображення. Колір, який сприймається людським оком, насправді може бути визначений трьома числами. Три основні кольори можуть створити досить велику частку набору сприйманих кольорів. Кольорові моделі зображення використовуються для представлення кольорів у цифровому вигляді за допомогою каналів [4]. Канали – це зображення у градаціях сірого, які містять інформацію різного типу. Три найпоширеніші моделі – RGB, CMYK та HSV. Наприклад, зображення RGB повинно мати канал для кожного кольору (червоного, зеленого та синього), а HSV має канали тону, насиченості і освітлення. Зображення можна розділити на канали RGB, HSL або CMYK незалежно від того, яка кольорова модель обрана для відображення. На рисунку 1.1 можна побачити, що крайнє праве зображення розділене на червоний, зелений та синій канали.



Рисунок 1.1 – Зображення, розділене на 3 канали

## 1.2 Геометричні перетворення

Геометричні перетворення зображень – це процес зміни форми, розміру, орієнтації та розташування зображення. Вони можуть використовуватися для виправлення спотворень перспективи, повороту зображення, зміни масштабу та обрізання зображення.

Якщо представити точки зображення в однорідних координатах, то всі перетворення можна реалізувати шляхом множення на матрицю, що значно прискорює загальний час виконання перетворень [5].

Масштабування є зміною розміру зображення відповідно до заданого масштабного коефіцієнта. Цей процес може виконуватись як зменшенням, так і збільшенням зображення. При зменшенні зображення його розміри зменшуються з використанням різних методів зменшення, що дозволяють зберігати якість зображення. При збільшенні зображення змінюються розміри зображення з більшою кількістю пікселів, ніж на початку. Це може бути корисно, якщо необхідно детальніше роздивитися певну частину зображення або надрукувати зображення в більшому форматі. Оскільки при збільшенні зображення додаються нові пікселі, які не існували в оригінальному зображенні, то це може призвести до розмиття чи розмазування деталей, що може знизити якість зображення. Тому, важливо обрати відповідний метод збільшення та правильно налаштувати параметри збільшення для збереження максимальної якості зображення. Представивши точки зображення в однорідних координатах, можемо змінити його масштаб помноживши на наступну матрицю:

$$D = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

де  $D$  – матриця масштабування;

$s_x$  – коефіцієнт масштабування вздовж осі  $x$ ;

$s_y$  – коефіцієнт масштабування вздовж осі  $y$ .

Обертання зображення на певний кут щодо центру зображення або будь-якої іншої точки називають поворотом. Цей процес може бути корисним, коли нам потрібно змінити орієнтацію зображення, щоб воно було легше розглядати або відобразити на екрані відповідно до певних вимог. Під час повороту зображення кожен піксель зображення зберігає свої координати, але його нове місце на зображенні залежить від кута повороту. Наприклад, при повороті на 90 градусів всі рядки зображення стають стовпцями, а всі стовпці – рядками. Під час повороту зображення можуть виникати певні проблеми, такі як відсутність частини зображення після повороту або зменшення розміру зображення. Це зазвичай відбувається, коли зображення повертається на кут, який не є кратним 90 градусам. Щоб уникнути цих проблем, можна використовувати різні методи повороту зображення, які дозволяють зберігати розмір зображення та заповнювати пропущені частини зображення. Наприклад, при повороті зображення можна використовувати методи афінного перетворення, інтерполяції зображення або розтягування зображення, які дозволяють зберігати якість зображення після повороту. Поворот зображення здійснюється множенням точок зображення на матрицю:

$$R = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

де  $R$  – матриця повороту;

$\alpha$  – кут повороту зображення.

Перенесення – це процес переміщення всього зображення або його частини в певному напрямку на певну відстань. Цей процес може бути корисним, коли нам потрібно змістити зображення, щоб зосередитися на певній частині зображення або змінити його композицію. Під час перенесення зображення всі пікселі на зображенні зберігають свої координати, але їхнє нове місце залежить від відстані та напрямку переміщення. Наприклад, якщо

перенести зображення вправо на 50 пікселів, кожен піксель зі зображення буде зсунутий вправо на 50 пікселів від своєї початкової позиції. Під час перенесення зображення можуть виникати певні проблеми, такі як відсутність частини зображення після переміщення або зменшення розміру зображення. Це зазвичай відбувається, коли зображення переноситься за межі його початкового розміру. Щоб уникнути цих проблем, можна використовувати різні методи перенесення зображення, які дозволяють зберігати розмір зображення та заповнювати пропущені частини зображення. Наприклад, при перенесенні зображення можна використовувати методи звичайного зміщення, зміщення з заповненням пікселями або методи, що дозволяють масштабувати та збільшувати розмір зображення, щоб заповнити пропущені частини зображення. Перенести зображення можна помноживши пікселі зображення на матрицю:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix},$$

де  $T$  – матриця перенесення;

$t_x$  – точка зміщення зображення вздовж осі  $x$ ;

$t_y$  – точка зміщення вздовж осі  $y$ .

Обрізання зображення – це процес видалення частини зображення, щоб отримати більш маленьке або зосереджене зображення з бажаним складом та композицією. Під час обрізання зображення видаляються пікселі зображення, які не входять у вибрану область обрізки. Цей процес може бути корисним для зміни пропорцій зображення, вирівнювання композиції та зосередження на певній частині зображення. При обрізанні зображення важливо вибрати правильну область обрізки. Зазвичай це залежить від того, яку частину зображення ви хочете зберегти та на яких параметрах зображення ви хочете зосередитися. Деякі методи обрізання зображення дозволяють зберігати пропорції зображення, щоб запобігти виникненню спотворень чи зменшення

якості зображення. Після обрізання зображення може виникати проблема зі збільшенням шуму або втратою деталей. Це може статися, якщо обрізати занадто багато зображення або виберемо область зображення з низькою якістю. Щоб уникнути цих проблем, необхідно ретельно вибирати область обрізки та зберігати відповідну якість зображення.

Іноді об'єкт на фотографії може виглядати інакше, ніж у реальному світі через спотворення перспективи. Ці спотворення можуть виникнути внаслідок зйомки об'єкта з різної відстані та під різними кутами, і впливають на сприйняття реальних пропорцій та форм об'єкта на фотографії. Цю проблему вирішує контроль перспективи. Шляхом зміни кута огляду або розтягнення зображення, цей процес дозволяє змінити форму зображення, щоб забезпечити правильні пропорції та форму об'єкта на фотографії. Контроль перспективи є важливим інструментом для фотографів, архітекторів, дизайнерів та інших фахівців, які працюють з візуальними зображеннями та потребують правильної перспективи для досягнення бажаного ефекту.

Правильне сприйняття фотографії залежить від того, як розташовані на ній об'єкти. Якщо зображення дзеркально відобразити, то воно зміниться і може виглядати зовсім інакше. Ефект дзеркального відображення можна використовувати для створення композиції та доповнення зображень одне одним. Дзеркально відображати фото можна як горизонтально, так і вертикально. Крім того, цей ефект може бути корисним для виправлення деяких недоліків на фотографії, наприклад, щоб прибрати непотрібні написи, символи або дзеркально спрямовані стрілки, які можуть плутати або призводити до помилок сприйняття.

### 1.3 Фільтрація зображень

Зображення зазвичай містять не тільки корисну інформацію, а ще й різні перешкоди. В таких випадках на допомогу приходять фільтри. Фільтр –

операція, що застосовується до зображення для зміни його зовнішнього вигляду або отримання певної інформації. Зазвичай фільтри застосовуються після зйомки, щоб надати зображенню бажаний ефект і використовуються для видалення шуму, розмиття або підвищення різкості зображення. Всі види фільтрів можна розділити на такі класи: частотні, лінійні, нелінійні, комбіновані [6].

Лінійні фільтри є найбільш поширеними і дають хороші результати при гауссовому адитивному шумі. Вони використовують маску фільтра. Лінійний фільтр є математичною операцією, яка обчислює нове значення пікселя на основі значень пікселів, що оточують його в оброблюваному зображенні. Іншими словами, маска фільтра є важливим інструментом для реалізації різних фільтраційних операцій. Вона використовується як шаблон для обробки кожного пікселя вхідного зображення. Маска фільтра містить числа, які відповідають коефіцієнтам фільтра, який буде застосовуватися до кожного пікселя зображення. Маска фільтра може бути різного розміру та форми в залежності від типу фільтрації, яку необхідно застосувати. Наприклад, маска розміром  $3 \times 3$  буде представлена у вигляді прямокутної матриці:

$$H = \frac{1}{K} * \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

де  $H$  – маска;

$a_{kl}$  – коефіцієнти маски;

$K$  – нормуючий множник, що дорівнює сумі усіх вагових множників.

Матриця розміром  $3 \times 3$  містить три рядки по три числа в кожному. Щоб перетворити один піксель в зображенні, значення його яскравості множиться на число в центрі маски. Потім проводиться множення восьми значень яскравості пікселів, що оточують центральний піксель, на відповідні їм коефіцієнти ядра, всі дев'ять значень підсумовуються, у результаті виходить

нове значення яскравості центрального пікселя. Цей процес повторюється для кожного пікселя у зображенні, тим самим зображення фільтрується.

## 1.4 Сучасні застосунки для обробки фото

Існує безліч програм для обробки зображень, які надають широкий набір інструментів для роботи з фотографіями, малюнками та іншими видами зображень. У цьому підрозділі буде розглянуто кілька найбільш популярних застосунків.

### 1.4.1 Adobe Photoshop

Adobe Photoshop – це найбільш відомий редактор зображень, який надає безліч інструментів та функцій для створення, редагування та покращення зображень, включаючи складні функції, такі як багатошарова обробка, ретушування, налаштування кольору та багато інших, що дозволяє користувачеві створювати складні композиції та покращувати якість зображень [7]. Він може бути використаний для роботи з фотографіями, малюнками та іншими видами зображень. Крім того, Adobe Photoshop також має велику спільноту користувачів, що дозволяє швидко знайти відповіді на будь-які питання, пов'язані з використанням програми. Однак, через складність і вартість, це може бути не найкращий вибір для новачків або для тих, хто шукає простий і швидкий спосіб покращення своїх фотографій. Adobe Photoshop може бути ідеальним інструментом для професійних фотографів і дизайнерів, але для новачків або людей, які шукають швидкий і простий спосіб покращення своїх фотографій, більш прості та доступні програми можуть бути кращим вибором.

### 1.4.2 GIMP

GNU Image Manipulation Program или GIMP – це безкоштовний кросплатформенний редактор зображень, доступний для GNU/Linux, macOS, Windows та інших операційних систем [8]. Застосунок є потужним та функціональним редактором, який може бути чудовою альтернативою для Adobe Photoshop для користувачів, які не хочуть витратити гроші на ліцензію, але програма може не мати деяких функцій, які є у Photoshop, що може затруднити роботу професіоналам у деяких областях, таких як дизайн та робота з векторними зображеннями.

Загалом, GIMP є потужним та корисним редактором зображень, який може задовольнити потреби більшості користувачів. Однак, перед використанням програми, користувачам, які не мають досвіду у використанні графічних редакторів, може знадобитися деякий час на вивчення її функцій та інтерфейсу.

### 1.4.3 Google Photos

Google Photos – це програма більше призначена для зберігання та організації фотографій та відео, але також має вбудовані інструменти для редагування зображень [9]. Застосунок доступний на різних платформах, включаючи Android, iOS та вебверсію. Google Photos надає користувачеві багато функцій для покращення та редагування фотографій, наприклад: корекція кольору, обрізка та поворот, видалення ефекту «червоних очей», фільтри, корекція портретів, додавання тексту тощо.

Крім того, Google Photos має зручний інтерфейс і простий у використанні, що робить його доступним для користувачів із будь-яким рівнем досвіду у редагуванні фотографій, але не варто очікувати від редактора такого діапазону інструментів, як в Adobe Photoshop.

## 1.5 Постановка задачі

Методи та алгоритми обробки фотореалістичних зображень є актуальними на сьогоднішній день, бо допомагають вирішувати реальні проблеми у багатьох галузях, де використовуються зображення, зокрема в медицині, технологіях, мистецтві, науці та багатьох інших. Тому ставиться задача розробити застосунок для обробки фотореалістичних зображень з урахуванням можливостей сучасних технологій та вимог користувачів.

Об'єктом роботи є реалізація застосунку для обробки фотореалістичних зображень.

Метою роботи є розробка застосунку зі зручним і інтуїтивно зрозумілим інтерфейсом, що базується на використанні сучасних методів і алгоритмів обробки фотореалістичних зображень, який забезпечить високу якість обробки зображень.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз та вибір сучасних методів та алгоритмів обробки фотореалістичних зображень;
- розробити функції, що дозволять користувачам здійснювати редагування зображень, такі як геометричні перетворення, фільтри тощо;
- розробити інтуїтивно зрозумілий та зручний інтерфейс користувача для застосунку, що дозволить користувачам легко використовувати його та отримувати бажаний результат;
- провести тестування та оптимізацію застосунку, щоб забезпечити високу швидкість обробки зображень та мінімальне споживання ресурсів.

## 2 АЛГОРИТМИ ОБРОБКИ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ

### 2.1 Видалення шумів

Шум на цифрових зображеннях може бути результатом електронних перешкод у процесі зйомки або може бути доданий в процесі компресії та збереження зображення. Існують різні типи шуму, такі як гаусівський шум, шум «сірого» каналу, шум смужок, шум плям та інші, які вимагають різних підходів до видалення.

Шум може негативно впливати на якість зображення, знижуючи його роздільну здатність, контрастність, насиченість кольорів та інші параметри. Видалення шуму, викликаного, наприклад, поганими умовами освітлення, може також допомогти покращити якість зображення та результати роботи алгоритмів пошуку КТ, наприклад, FAST, BRIEF, ORB [10–13].

#### 2.1.1 Фільтр Гауса

Для видалення шумів із зображення можна використовувати різні методи та алгоритми, залежно від типу шуму та характеристик зображення. Один із найпростіших методів – це застосування фільтра Гауса. Фільтр Гауса розмиває зображення, видаляючи шум, зберігаючи основні контури і форми зображення. Часто використовується симетричне гаусове ядро як формальна модель згортки для «розмивання» зображень у вигляді [14, 15]

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

де  $H(x, y)$  – значення елемента матриці з координатами  $x, y$ ;

$\sigma$  – середньоквадратичне відхилення гаусіану в пікселях.

У ядрі фільтра Гауса центральний елемент має найбільше значення, а пікселів зменшується в міру віддалення від центру. Стандартне відхилення функції Гауса визначає, наскільки швидко зменшуються значення ядра з віддаленням від центру. Більше значення стандартного відхилення призводить до більш розмитого зображення.

Для дискретного випадку ядро згладжування має вигляд масиву розміром  $(2k+1) \times (2k+1)$ , значення елемента  $(i, j)$  якого дорівнює

$$H_{ij} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2}},$$

де  $k$  – радіус ядра фільтрації.

Наприклад, ядро згладжування  $3 \times 3$  з середньоквадратичним відхиленням  $\sigma=1$  буде виглядати наступним чином:

$$\begin{bmatrix} 0,05855 & 0,096532 & 0,05855 \\ 0,096532 & 0,159155 & 0,096532 \\ 0,05855 & 0,096532 & 0,05855 \end{bmatrix}.$$

Зображення з шумом показано на рисунку 2.1 а). Якщо середньоквадратичне відхилення гаусіану невелике, то згладжування дає незначний результат, так як вагові коефіцієнти центральних пікселів дуже малі, наприклад, як показано на рисунку 2.1 б).

Коли значення середньоквадратичного відхилення велике, це означає, що пікселі в околі мають значення, які досить відрізняються одне від одного. У такому випадку вагові коефіцієнти для цих пікселів будуть більшими, що призведе до більшої впливу цих пікселів на середнє значення. Це допоможе зменшити шум, оскільки значення пікселів, які мають більші вагові коефіцієнти, будуть більш вагомими при обчисленні середнього значення. Це показано на рисунку 2.1 в).

Але велике значення середньоквадратичного відхилення може також призвести до видалення більшої частини елементів зображення, що може призвести до втрати деталей та текстур в зображенні, що продемонстровано на рисунку 2.1 г) [5].

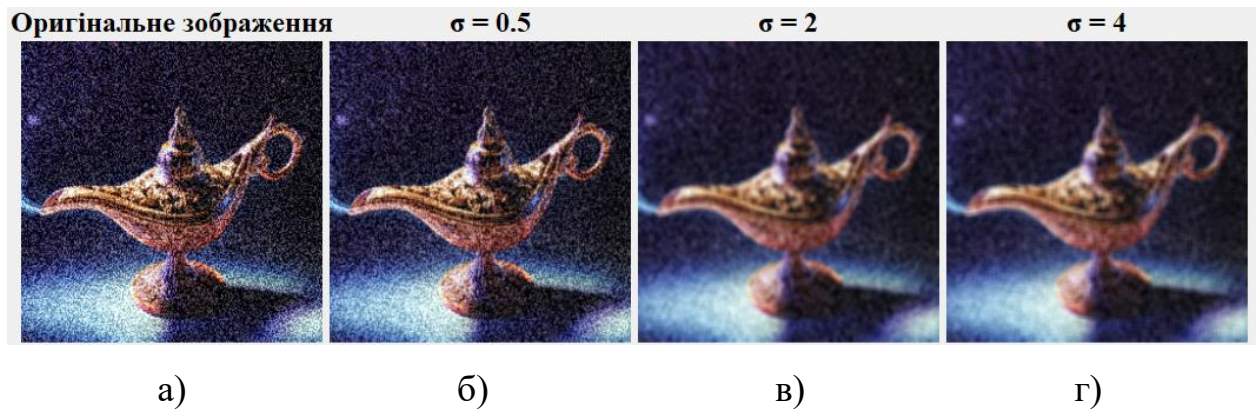


Рисунок 2.1 – Результат усунення шуму за допомогою фільтра Гауса:  
 а) оригінальне зображення; б) середньоквадратичне відхилення дорівнює 0,5;  
 в) середньоквадратичне відхилення дорівнює 2; г) середньоквадратичне відхилення дорівнює 4

### 2.1.2 Медіанний фільтр

Ще один метод – це медіанний фільтр. Він видаляє шум, замінюючи кожен піксель на медіану яскравості пікселів у невеликому околі. Медіанний фільтр добре працює для видалення імпульсного шуму, який з'являється випадково на зображенні. Його використовують для зменшення «шуму солі та перцю» і для видалення подряпин на фото.

Застосування медіанного фільтра до зображення зводиться до заміни значення кожного вхідного пікселя на медіану значень яскравості пікселів, що знаходяться в околі даного пікселя.

Нехай, наприклад, використовується маска  $3 \times 3$  на зображенні, де пікселі в околі мають більш-менш однакове значення, але в центрі маски є окремий

піксель з відхиленням від цього значення, то його можна вважати шумовим викидом.

$$\begin{bmatrix} 135 & 132 & 140 \\ 160 & 10 & 153 \\ 145 & 149 & 153 \end{bmatrix}.$$

Варіаційний ряд має вигляд 10, 132, 135, 140, 145, 149, 153, 153, 160. У даному випадку медіанне значення – 145, бо маска  $3 \times 3$ , тобто 9 елементів, значить медіанний елемент буде 5 в порядку зростання. Якби була маска  $5 \times 5$  або  $7 \times 7$ , то медіанним елементом був би 13 або 25 елемент відповідно. Після того як медіанне значення знайдено, елемент, який містить шумовий викид замінюється на нього.

Медіанну фільтрацію можна здійснити за наступною схемою:

Крок 1. Як і у більшості підходів необхідно проаналізувати локальний окіл у вигляді матриці  $n \times n$  точок зображення. Як правило, параметр  $n$  вибирають непарним [16].

Крок 2. Розмістити маску на кожному пікселі вхідного зображення.

Крок 3. Зібрати всі значення яскравості пікселів у фільтрі в один список.

Крок 4. Відсортувати список значень по зростанню.

Крок 5. Замінити значення поточного пікселя на медіану відсортованого списку значень.

Нехай на вхід отримано зображення з «шумом солі та перцю», яке показано на рисунку 2.2 (а).

Якщо розмір маски фільтра занадто малий, то він може пропустити деякі шуми, які можуть вплинути на якість зображення. Цей недолік показано на рисунку 2.2 (б).

Навпаки, якщо розмір маски фільтра занадто великий, то це може призвести до змазаного зображення, оскільки медіана буде взята занадто багатьох значень, в тому числі й з значеннями поза реальним діапазоном яскравості пікселів, що продемонстровано на рисунку 2.2 (в).



(a)

(б)

(в)

Рисунок 2.2 – Результат усунення шуму за допомогою медіанного фільтра:  
 (а) оригінальне зображення; (б) розмір маски  $3 \times 3$  0,5; (в) розмір маски  $27 \times 27$

Недоліком є те, що медіанний фільтр може бути дуже повільним на великих зображеннях, оскільки він вимагає сортування значень пікселів у ядрі фільтра кожного пікселя зображення і, хоча, він частіше за все видаляє деякі дрібні деталі зображення, які є частиною шуму, але вони можуть бути важливими частинами зображення.

### 2.1.3 Фільтрація за середнім значенням

Фільтрація за середнім значенням є простим методом видалення шуму. Він працює шляхом заміни значення кожного пікселя у зображенні на середнє значення яскравості пікселів навколо нього. Як ядро можна використовувати квадратну матрицю певного розміру (наприклад,  $3 \times 3$ ,  $5 \times 5$  тощо), яка ковзає по всіх пікселях зображення. Значення кожного пікселя замінюється на середнє значення всіх пікселів у ядрі. Цей процес виконується кожного пікселя, що призводить до згладжування шуму. Однак цей метод може викликати розмиття деталей та країв зображення, оскільки він не враховує різну структуру та текстуру у зображенні.

Наприклад, ядро згладжування для фільтра за середнім значенням  $5 \times 5$  буде виглядати наступним чином:

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Фільтрація за середнім значенням застосовується в різних областях комп'ютерного зору та обробки зображень, де необхідно видалити шум для покращення якості зображення. Він є простим та швидким методом, але може бути недостатнім для складних зображень із різноманітними текстурами та деталями. У таких випадках може бути корисно розглянути фільтрацію Гауса або медіанну фільтрацію. Результат дії видалення шуму за допомогою фільтрації за середнім значенням продемонстровано на рисунку 2.3.



Рисунок 2.3 – Результат видалення шуму за допомогою фільтра за середнім значенням

#### 2.1.4 Комбінація різних методів обробки

Для визначення найкращого методу видалення шумів для конкретного зображення необхідно враховувати тип шуму, його інтенсивність та характеристики зображення, а також вимоги до якості вихідного зображення.

Найкращий результат видалення шуму з фотореалістичних зображень можна отримати, використовуючи комбінацію різних методів обробки. Кожен

метод видалення шуму має свої переваги та обмеження, тому комбінація кількох методів може допомогти покращити якість зображення та зменшити візуальний шум. Комбінація різних методів може бути більш ефективною, ніж використання будь-якого з них окремо, оскільки це дозволяє видалити різні типи шуму та зберегти якість зображення. Як було сказано раніше, відповідний вибір методу видалення шуму залежить від характеристик зображення та типу шуму.

Для покращення якості зображення можна використовувати не тільки методи видалення шуму, але й інші методи обробки, такі як корекція контрастності, збільшення роздільної здатності, зменшення спотворень та інші. Комбінація різних методів може допомогти досягти найкращих результатів в обробці фотореалістичних зображень.

## 2.2 Колірна корекція

Колірна корекція – це процес виправлення, покращення або зміни кольорів в зображенні з метою досягнення бажаного візуального ефекту. Вона може бути застосована до фотографій, відео, графічних зображень або інших типів медіа. Основна мета колірної корекції – досягнення точності відтворення кольорів, що відповідають оригіналу або заданому естетичному стилю.

Корекція яскравості та контрастності є параметрами, які впливають на загальний рівень освітленості зображення (яскравість) і різницю між світлими і темними областями (контрастність). Зміна цих параметрів може зробити зображення більш насиченим, деталізованим або, навпаки, більш приглушеним.

Необхідно пам'ятати, що надмірне збільшення яскравості або контрастності може призвести до втрати деталей та появи шуму на зображенні. Тому рекомендується помірно використовувати ці налаштування та уважно стежити за результатом.

Методи корекції кольору використовуються для налаштування та покращення колірної балансу на зображенні. Різні методи корекції кольору дозволяють внести зміни у відтінки, насиченість і яскравість кольорів, щоб досягти бажаного візуального ефекту. Іноді колірні дані на зображенні є важливим фактором для класифікації зображень, тому колірні дані можуть бути корисними навіть в таких випадках [17–21].

### 2.2.1 Алгоритм «сірого світу»

Один з найпоширеніших методів корекції кольору – це алгоритм «сірого світу». Цей підхід ґрунтується на припущенні, що середній колір на зображенні має бути сірим. Використовуючи це припущення, можна зробити корекцію кольорів, що приводить кольори в більш збалансований стан. Наприклад, якщо зображення має сильний синюватий відтінок, застосування алгоритму «сірого світу» дозволить прибрати цей відтінок і досягти більш природного сприйняття кольору.

Алгоритм «сірого світу» можна здійснити за наступною схемою:

Крок 1. Визначити середнє значення яскравості (середню яскравість пікселів) для кожного каналу кольору (червоного, зеленого та синього) у зображенні. Наприклад, для червоного каналу середнє значення яскравості можна розрахувати наступним чином:

$$R_{cp} = \frac{1}{a * b} * \sum_{i=1}^a \sum_{j=1}^b r_{ij},$$

де  $R_{cp}$  – середнє значення яскравості червоного каналу;

$a$  – ширина зображення;

$b$  – висота зображення;

$r_{ij}$  – значення яскравості червоного каналу пікселя з координатами  $ij$ .

Крок 2. Обчислити середнє значення яскравості всього зображення шляхом усереднення значень кожного каналу:

$$I = \frac{R_{\text{cp}} + G_{\text{cp}} + B_{\text{cp}}}{3}.$$

Крок 3. Розрахувати коефіцієнти корекції кожного каналу шляхом поділу середнього значення яскравості всього зображення на середнє значення яскравості кожному за каналу. Наприклад, для червоного каналу коефіцієнт корекції розраховується наступним чином:

$$k_R = \frac{I}{R_{\text{cp}}}.$$

Коефіцієнти коригування використовуються для масштабування значень пікселів кожного каналу.

Крок 4. Помножити кожен піксель зображення на відповідний коефіцієнт корекції його каналу. Це призведе до балансування кольорів та усунення колірної зміщення.

Результатом алгоритму «сірого світу» буде збалансоване кольорове зображення, у якому середній рівень яскравості відповідатиме середньому рівню яскравості сірого зображення. Це допомагає усунути небажані відхилення кольорів і створити більш природний зовнішній вигляд зображення. Загалом, алгоритм «сірого світу» дозволяє досягти більш природного і збалансованого візуального представлення кольорів у зображенні, покращуючи його зовнішній вигляд і якість.

Нехай на вхід дано зображення з синюватим відтінком показане на рисунку 2.4 (а). Застосувавши до нього корекцію кольору за допомогою алгоритму «сірого світу», отримаємо зображення зі збалансованими каналами, що показано на рисунку 2.4 (б).



(a)

(б)

Рисунок 2.4 – Результат корекції кольору за допомогою алгоритму «сірого світу»: (a) оригінальне зображення; (б) збалансоване зображення

Алгоритм «сірого світу» простий у реалізації і не вимагає складних обчислень чи складних алгоритмічних методів. Він автоматично визначає зсув кольору і самостійно коригує баланс кольору, що дозволяє отримати більш природні кольори в зображенні, але він може не давати хороших результатів у випадку, коли є велика кількість різних джерел світла або яскравих об'єктів, а також за наявності сильного контрасту та насиченості кольорів. Іноді може відбуватися втрата інформації про кольори в зображенні, оскільки алгоритм прагне балансу і нейтрального відтінку кольору, що може бути небажано в деяких ситуаціях, наприклад, у фотографії з особливим емоційним відтінком або специфічною атмосферою.

### 2.2.2 Автоматичний баланс білого

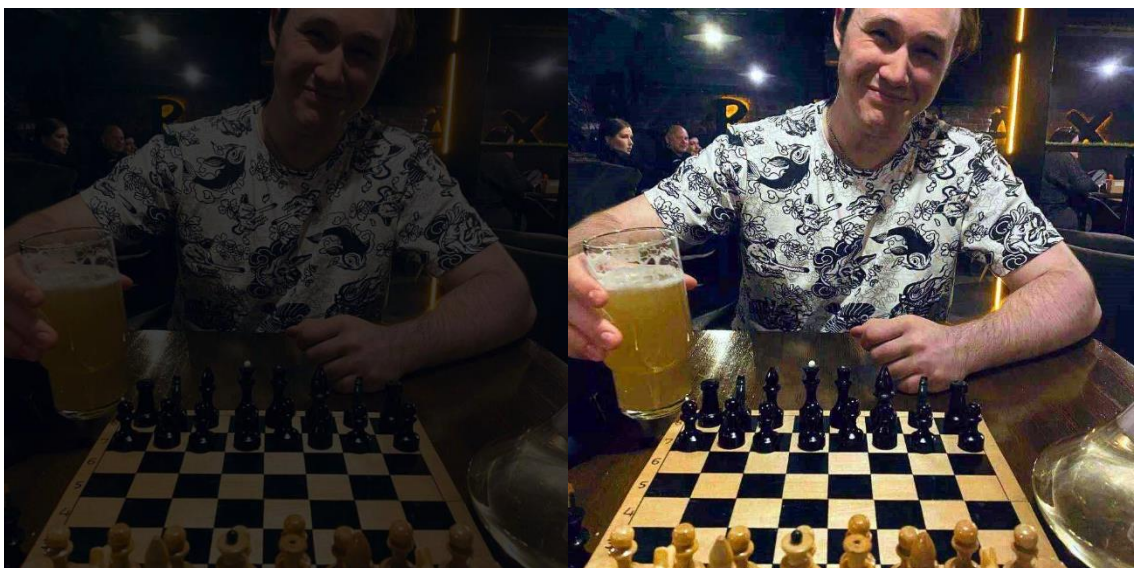
Ще один поширений метод – автоматичний баланс білого. Цей метод спрямований на корекцію відтінків, пов'язаних із висвітленням сцени. Він використовується для усунення небажаних відтінків кольорів, викликаних різними джерелами світла, такими як лампи різного типу або природне освітлення в різний час доби. Баланс білого дозволяє узгодити колірну гаму з

відомим джерелом світла, щоб досягти більш точної передачі кольору на фотографії [22].

У ситуаціях, коли джерело світла має різну температуру кольору (наприклад, денне світло, освітлення лампою розжарювання, флуоресцентні лампи тощо), кольорні відтінки зображення можуть бути зміщеними. Автоматичне балансування білого вирішує цю проблему шляхом автоматичного коригування кольорного балансу відповідно до умов освітлення.

Принцип роботи автоматичного балансування білого полягає у визначенні «білої точки» у кадрі, яка є об'єктом, який повинен бути білим або мати нейтральний колір. Камера аналізує відтінки кольорів, визначає їх відхилення від нейтрального білого кольору та автоматично коригує баланс кольорів, щоб досягти більш правильного відображення кольорів на зображенні.

Нехай на вхід дано зображення з поганим освітленням показане на рисунку 2.5 (а). Застосувавши до нього корекцію кольору за допомогою балансу білого, отримаємо зображення, показане на рисунку 2.5 (б), яке точно відображає колір.



(а)

(б)

Рисунок 2.5 – Результат корекції кольору за допомогою балансу білого:

(а) оригінальне зображення; (б) збалансоване зображення

Баланс білого є важливим інструментом обробки кольору, який дозволяє досягти точного відображення кольорів на зображенні. Розуміння різних методів корекції балансу білого та їх застосування залежно від конкретних умов освітлення допоможе створювати якісні та реалістичні зображення.

### 2.3 Обрізка та зміна розміру

Обрізка та зміна розміру зображень є важливими алгоритмами обробки фотореалістичних зображень. Ці алгоритми дозволяють змінювати розмір та область зображення з метою покращення композиції, видалення непотрібних деталей або пристосування зображення до певних вимог.

Обрізка зображень дозволяє вибрати певну область зображення, видаляючи непотрібні частини. Це може бути корисно для поліпшення композиції зображення, виключення відволікаючих елементів або зосередження на конкретному об'єкті. Алгоритми обрізки зображень можуть враховувати різні фактори, такі як пропорції сторін зображення, відношення сторін області обрізки, збереження деталей та інші.

Зміна розмірів зображення може бути корисною для адаптації зображення до певних вимог, наприклад, для пристосування до розмірів вебсторінки, друку, екрану пристрою тощо. При зміні розмірів зображення важливо зберегти його якість і деталі, щоб уникнути розмиття або спотворення.

Ці процедури є стандартними операціями в багатьох програмах для редагування фотографій та графічних редакторах. Вони також можуть бути реалізовані за допомогою алгоритмів обробки зображень, які враховують різні аспекти, такі як збереження деталей, композиції та інші фактори для досягнення бажаного результату.

### 2.3.1 Обрізка

Обрізка дозволяє користувачеві вибрати область зображення, яку потрібно зберегти, а решту видалити. Це дозволяє точно контролювати композицію та зберегти лише потрібну частину зображення.

Якщо користувач хоче зосередитися на певному об'єкті або деталі фотографії, обрізка дозволяє йому видалити зайві елементи, що відволікають увагу, та збільшити значущість вибраного об'єкта. Також обрізка може допомогти покращити загальну композицію фотографії, змінюючи її пропорції або використовуючи золотий перетин, правило третин або інші композиційні принципи. Якщо на вашій фотографії є непотрібні або завадливі елементи, такі як люди, маркери або нежадані об'єкти, цей інструмент може бути використаний для видалення та створення більш чистого і естетичного зображення.

Обрізають фото зазвичай покроково:

Крок 1. Вибір області обрізки. Користувач визначає область на зображенні, яку він бажає зберегти. Це може бути встановлення прямокутного або круглого ділянку, або вручну маркуючи область.

Крок 2. Перевірка валідності області. Перевіряється, чи обрана область знаходиться в межах зображення і не виходить за його межі. Якщо обрана область недійсна, можуть застосовуватися певні корекції, такі як автоматичне масштабування або збільшення розміру зображення.

Крок 3. Обрізка зображення. Зображення вирізається за межами обраної області. Всі пікселі, що не належать до обраної області, видаляються.

Крок 4. Збереження обрізаного зображення. Обрізане зображення може бути збережене у вибраному форматі (наприклад, JPEG, PNG або TIFF) з певними налаштуваннями якості, розміру або іншими параметрами.

Нехай на вхід дано зображення з великою кількістю зайвих деталей, яке показано на рисунку 2.6 (а). Застосувавши до нього обрізку, отримаємо

зображення, показане на рисунку 2.6 (б), яке містить лише потрібні користувачу деталі.



(а)

(б)

Рисунок 2.6 – Результат роботи алгоритму обрізки:  
(а) оригінальне зображення; (б) обрізане зображення

Обрізка є потужним інструментом в обробці фотореалістичних зображень, який дозволяє покращити їх композицію, фокус, естетику та пристосувати до конкретних вимог або використання.

### 2.3.2 Масштабування зображення

Масштабування зображення стосується процесу зміни розміру зображення, щоб зробити його більшим або меншим, зберігаючи його пропорції та візуальну якість.

Коли зображення збільшується, генеруються додаткові пікселі, щоб збільшити його розмір, тоді як зменшення передбачає видалення пікселів, щоб зменшити розміри. Мета масштабування зображення може бути різною, наприклад, підганяти зображення під певний розмір дисплея, зменшувати

розмір файлу для ефективного зберігання чи передачі або адаптувати зображення для різних пристроїв чи роздільної здатності.

Масштабування зображення може бути простою операцією, коли розмір кожного пікселя регулюється пропорційним, або може включати складніші алгоритми, які намагаються зберегти якість і деталі вихідного зображення. Алгоритми масштабування спрямовані на мінімізацію візуальних артефактів і спотворень, які можуть виникнути під час процесу зміни розміру, таких як розмиття, нерівні краї або згладжування.

Різні алгоритми масштабування використовують різні методи, такі як інтерполяція, фільтрація або складні математичні обчислення, щоб визначити нові значення пікселів під час зміни розміру зображення. Вибір алгоритму залежить від таких факторів, як бажана якість зображення, ефективність обчислення та специфічні характеристики зображення, яке масштабується. Найбільш відомі методи зміни розміру: інтерполяція найближчого сусіда, білінійна інтерполяція і бікубічна інтерполяція.

Інтерполяція найближчого сусіда – найпростіший алгоритм масштабування, який працює шляхом дублювання або відкидання пікселів для досягнення бажаного розміру. Це швидко, але може призвести до згладжування та втрати якості зображення. Наприклад, є зображення розміром 4×4 пікселів:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}.$$

Для збільшення цього зображення, наприклад, до розміру 8×8 пікселів застосуємо алгоритм інтерполяція найближчого сусіда. Для цього необхідно кожен піксель вихідного зображення скопіювати в нове зображення, і кожен піксель нового зображення отримає таке значення, як і найближчий піксель вихідного зображення:

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 \\ 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 \\ 9 & 9 & 10 & 10 & 11 & 11 & 12 & 12 \\ 9 & 9 & 10 & 10 & 11 & 11 & 12 & 12 \\ 13 & 13 & 14 & 14 & 15 & 15 & 16 & 16 \\ 13 & 13 & 14 & 14 & 15 & 15 & 16 & 16 \end{bmatrix}.$$

Нехай дано зображення на рисунку 2.7 (а). Необхідно збільшити його вдвічі. Якщо використати інтерполяцію найближчого сусіда отримаємо результат, зображений на рисунку 2.7 (б).

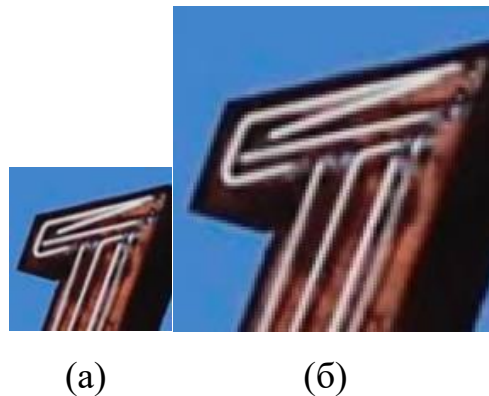


Рисунок 2.7 – Результат збільшення зображення за допомогою інтерполяції найближчого сусіда: (а) оригінальне зображення розміром 96×96 пікселів; (б) зображення збільшене вдвічі

Білінійна інтерполяція. Алгоритм бере середнє зважене значення найближчих чотирьох пікселів, щоб визначити значення нового пікселя. Він дає більш плавні результати порівняно з найближчим сусідом, але все одно може призвести до деякого розмиття. Наприклад, є зображення 2×2 пікселів:

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}.$$

Необхідно збільшити його вдвічі. Використавши білінійну інтерполяцію, отримаємо наступне:

$$\begin{bmatrix} 10 & 13.3 & 16.6 & 20 \\ 15 & 18.5 & 21.5 & 25 \\ 25 & 28.5 & 31.5 & 35 \\ 30 & 33.3 & 36.6 & 40 \end{bmatrix}.$$

Таким чином, з використанням білінійної інтерполяції розмір зображення збільшується до 4×4 пікселів, і нові значення пікселів обчислюються на основі навколишніх значень вихідного зображення, що призводить до більш плавного і точного збільшення розміру зображення.

Візьмемо те ж зображення, що і для інтерполяції найближчого сусіда і збільшимо його вдвічі за допомогою метода білінійної інтерполяції. Результат показано на рисунку 2.8.

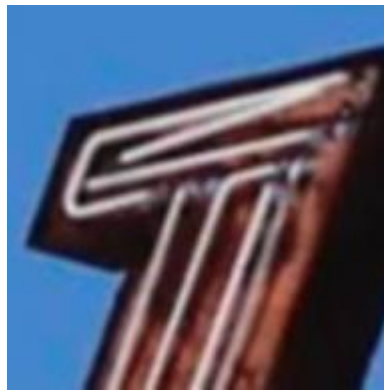


Рисунок 2.8 – Зображення збільшене вдвічі за допомогою білінійної інтерполяції

Бікубічна інтерполяція. Алгоритм працює за схожим принципом, що і білінійна інтерполяція і бере середнє зважене значення найближчих 16 пікселів для обчислення значення нового пікселя. Він дає результати вищої якості, ніж білінійна інтерполяція, з меншим розмиттям.

Загалом, вибір методу інтерполяції залежить від конкретної задачі та необхідної якості результату. Інтерполяція найближчого сусіда найпростіша, але може бути неприйнятною для завдань, які потребують точності та плавних переходів. Білінійна інтерполяція забезпечує більш гладкі результати, а бікубічна інтерполяція має найбільш точне і детальне відновлення даних.

### 3 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ ОБРОБКИ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ

#### 3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи був розроблений застосунок для обробки фотореалістичних зображень за допомогою алгоритмів фільтрації, алгоритмів інтерполяції, алгоритмів колірної корекції і геометричних перетворень. Для реалізації було обране середовище Microsoft Visual Studio 2022. Це обумовлено тим, що Microsoft Visual Studio 2022 є одним з популярних і потужних інтегрованих середовищ розробки (IDE) для програмування на мовах, таких як C++, C#, і Visual Basic. Він надає широкі можливості для створення програм, включаючи розробку десктопних застосунків, вебзастосунків і програм для мобільних пристроїв. Visual Studio має багато інструментів і функціональних можливостей, що допоможуть забезпечити швидку і продуктивну розробку зображення [23, 24].

Для розробки застосунка було обрано мову програмування C#. Це можна обґрунтувати тим, що мова C# є потужною і сучасною мовою програмування, яка добре підходить для розробки застосунків для обробки зображень. Вона має зручний синтаксис, підтримку об'єктно-орієнтованого програмування та широкий набір бібліотек для роботи з графікою та зображеннями. Використання мови C# дозволить зручно та ефективно реалізувати функціонал обробки зображень у застосунку.

Для роботи з фотореалістичними зображеннями в C# існує велика кількість потужних бібліотек. Наприклад, бібліотека AForge.NET надає широкий спектр функціональності для обробки та аналізу зображень, включаючи фільтрацію, видалення шумів, корекцію кольору тощо. Інші бібліотеки, такі як Emgu CV і Accord.NET, також пропонують потужні інструменти для обробки зображень. Використання Visual Studio 2022

дозволить легко інтегрувати ці бібліотеки у ваш застосунок та ефективно використовувати їх функціонал.

Операційна система Windows надає розроблювачам застосунків потужні засоби Інтерфейсу Графічних Пристроїв GDI (Graphics Device Interface) для побудови графічних зображень незалежно від типу використовуваного пристрою висновку [25]. На жаль, GDI обтяжує програмістів безліччю додаткових дій (зокрема, по керуванню системними ресурсами), які відволікають розроблювача від його основного завдання – створення графіки.

У середовищі розробки C# та платформі .NET розробникам надаються потужні засоби для створення графічних зображень, які звільняють їх від непродуктивної роботи з використанням GDI в Windows. Windows Forms є фреймворком для створення графічних застосунків на платформі .NET. Він надає простий інтерфейс програмування, що дозволяє легко створювати вікна, елементи керування, зображення та інші графічні компоненти. Windows Forms дозволяє розробникам швидко створювати інтерактивні застосунки з графічним інтерфейсом, що відповідає стандартам Windows [26, 27].

Windows Forms має наступні переваги:

- простота використання. Windows Forms надає простий та зрозумілий інтерфейс програмування для створення графічних застосунків. Можна швидко створювати вікна, елементи керування (кнопки, текстові поля, списки тощо) та маніпулювати ними з легкістю. Windows Forms дозволяє розробникам швидко створювати прості та функціональні інтерфейси користувача;

- широкий набір елементів керування. Windows Forms надає багато вбудованих елементів керування, які можна використовувати для створення різноманітних інтерфейсів. Можна легко додавати кнопки, тексти, випадаючі списки, таблиці та інші елементи керування до свого застосунку з використанням графічного інтерфейсу користувача;

- розширені можливості налаштування. Windows Forms дозволяє налаштовувати вигляд і поведінку елементів керування з використанням

властивостей та подій. Програміст може змінювати кольори, шрифти, розміри та інші властивості елементів керування за допомогою інтуїтивного інтерфейсу Microsoft Visual Studio. Крім того, можна визначати дії, які відбуваються при взаємодії з елементами керування, використовуючи події;

- підтримка стандартів Windows. Windows Forms розроблено для використання в операційній системі Windows, тому фреймворк повністю підтримує стандартні стилі та поведінку Windows. Застосунки, створені з використанням Windows Forms, будуть мати вигляд і відповідати стандартам Windows, що забезпечує зручність користування та знайомість для користувачів;

- доступність ресурсів .NET. Використання Windows Forms у середовищі .NET дозволяє використовувати широкий спектр функціональності, доступної в платформі .NET. Можна використовувати розширені бібліотеки для роботи з базами даних, мережевими операціями, обробкою зображень та багато іншого.

Інтерфейс створення застосунку Windows Forms за допомогою Microsoft Visual Studio показано на рисунку 3.1.

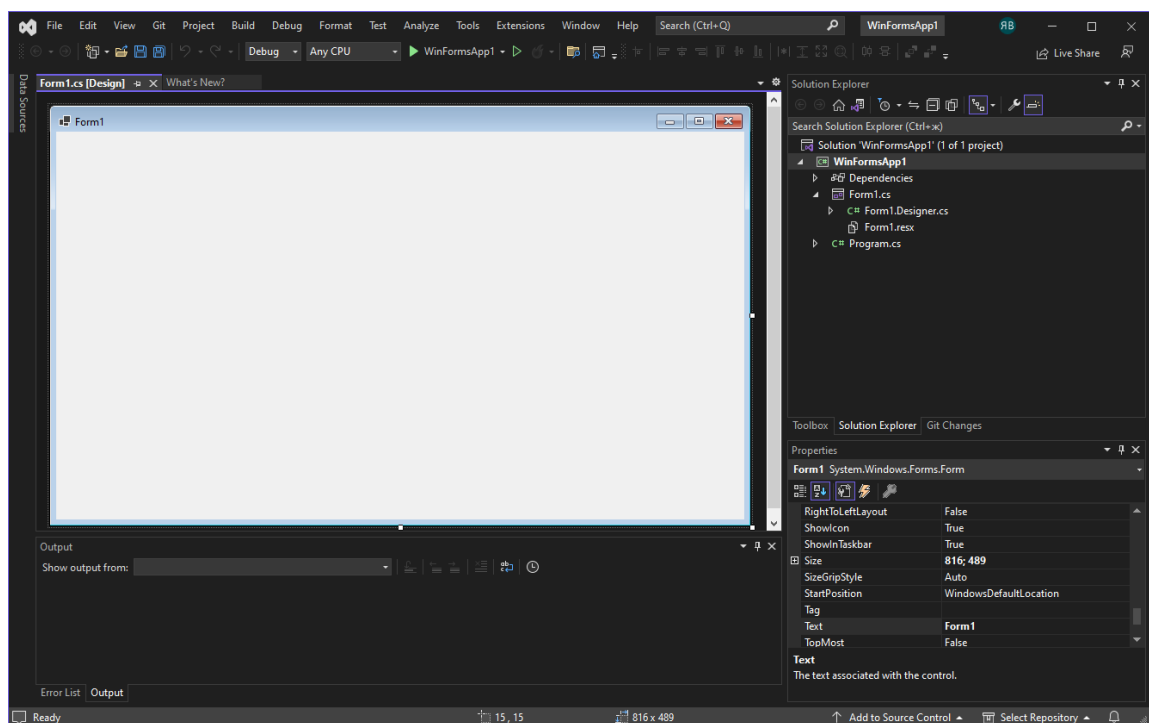


Рисунок 3.1 – Приклад інтерфейсу Windows Forms

Для обробки фотореалістичних зображень було обрано бібліотеку Emgu CV. Emgu Computer Vision є .NET обгорткою для відкритої бібліотеки комп'ютерного зору OpenCV. Бібліотека OpenCV містить понад 2500 оптимізованих класичних та сучасних алгоритмів аналізу зображень і машинного навчання [28–30]. Вона надає зручний доступ до функцій обробки зображень та комп'ютерного зору у мові C#.

Emgu CV розширює OpenCV, надаючи зручний API для взаємодії з функціями OpenCV з використанням C#. Вона забезпечує інтерфейс для завантаження та збереження зображень, виконання операцій обробки зображень, виявлення об'єктів, аналізу зображень та багато іншого. Крім того, Emgu CV надає додаткові функції, які полегшують використання OpenCV в C#. Вона надає обгортки навколо деяких складних функцій OpenCV, що робить їх більш доступними для розробників. Також Emgu CV забезпечує підтримку мультимедіа (зокрема роботу з відео та аудіо), що не входить в стандартний функціонал OpenCV.

Загалом, Emgu CV доповнює OpenCV, дозволяючи розробникам використовувати всю потужну функціональність OpenCV у своїх проєктах на C#, зберігаючи зручність та простоту програмування мови C#.

Вибір даної бібліотеки обґрунтовано тим, що вона дозволяє використовувати повний набір функцій OpenCV у застосунку на мові C#. За допомогою Emgu CV можна виконувати такі завдання, як розпізнавання облич, об'єктів та багато іншого.

Бібліотека забезпечує швидку обробку зображень завдяки використанню оптимізованих алгоритмів OpenCV та використанню мови C#. Вона дозволяє працювати з великими наборами даних та виконувати обробку зображень у реальному часі. EmguCV інтегрується з платформою .NET, що дозволяє використовувати її разом з іншими бібліотеками та фреймворками .NET. З'являється можливість поєднувати функціональність EmguCV з іншими інструментами для розробки графічних застосунків на мові C#.

EmguCV має детальну документацію та активну спільноту розробників, що допомагає вирішувати проблеми та отримувати підтримку. Розробники можуть знайти приклади коду, поради та розширену документацію для використання EmguCV у своїх проєктах.

### 3.2 Програмна реалізація

Розробка застосунку для обробки фотореалістичних зображень може бути розділена на наступні етапи:

- завантаження зображення. Необхідно реалізувати функцію, яка дозволяє користувачеві завантажити фото в застосунок. Це може включати інтерфейс користувача для вибору файлу або перетягування зображення в програму;

- попередня обробка. Застосування алгоритму колірної корекції «баланс білого», який виправляє несправжні кольори і забезпечує правильний баланс кольорів у зображенні. Встановлення оптимальних коефіцієнтів для алгоритму сірого світу, який регулює яскравість зображення і забезпечує зручний контраст;

- видалення шумів. Застосування фільтра Гауса, який розмиває зображення, зменшуючи високочастотні компоненти і видаляє шуми. Застосування фільтра за середнім значенням, який замінює кожний піксель у зображенні середнім значенням його сусідів, що допомагає згладити шуми. Застосування медіанного фільтра, який замінює кожний піксель у зображенні медіанним значенням його сусідів, що допомагає видалити шуми і зберегти різкі контури об'єктів;

- геометричні перетворення. Реалізовано алгоритми для зміни розміру, обертання, віддзеркалення та обрізки зображення. Обрізка, обертання і масштаб можуть включати вибір користувачем потрібного розміру або кута обертання, а також ручний вибір області обрізки;

– збереження зображення. Реалізовано функцію, яка дозволяє користувачеві зберегти оброблене зображення на комп'ютері після всіх обробок. Це може включати вибір користувачем назви файлу та формату зображення для збереження.

Інтерфейс розробленого застосунку показано на рисунку 3.2. На формі розташовано групові контейнери з відповідними підписами, які відповідають за обробку фотореалістичних зображень та елемент форми на якому буде відображатися зображення, завантажене користувачем. В групових контейнерах є кнопки, які викликають відповідні алгоритми обробки зображень. У деяких з них користувач має встановити певні значення в поля введення.

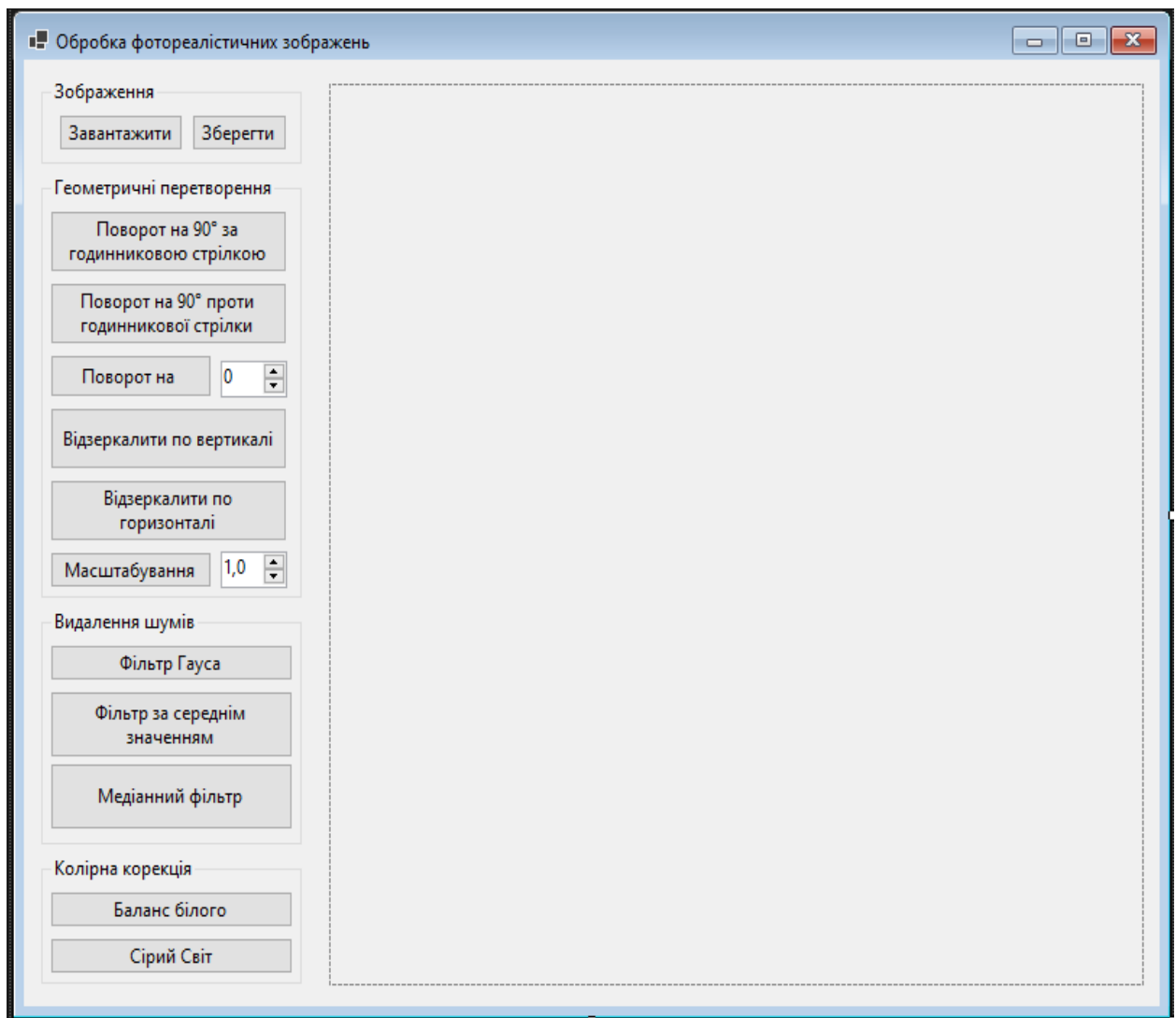


Рисунок 3.2 – Інтерфейс застосунку

Отже, при роботі з застосунком користувачеві, у першу чергу, необхідно завантажити зображення на форму, скориставшись кнопкою «Завантажити». Після завантаження зображення користувачу буде доступний різний функціонал для обробки фотографій.

Дані можливості дозволять користувачеві експериментувати з різними способами обробки фотографій, використовуючи розроблений функціонал застосунку.

### 3.3 Інструкція користувача

При запуску застосунку перед користувачем з'явиться вікно, яке зображене на рисунку 3.3.

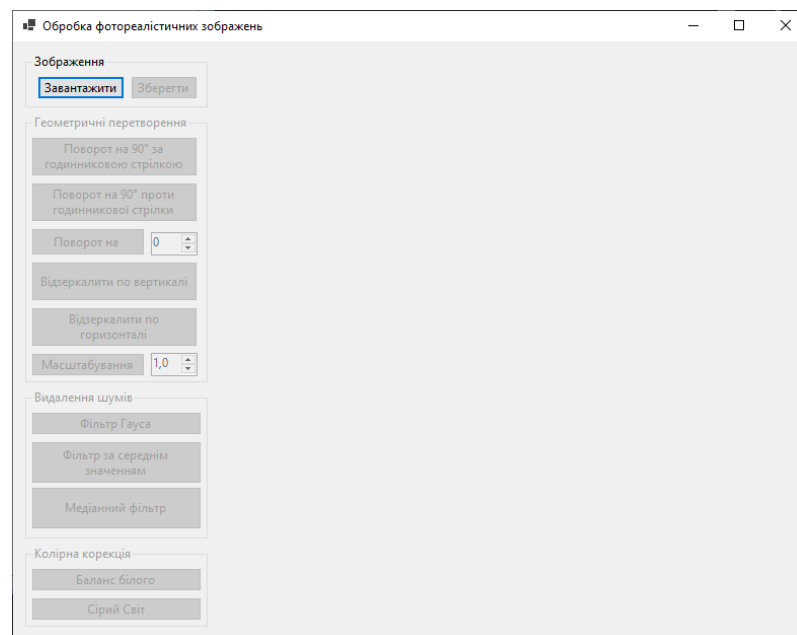


Рисунок 3.3 – Застосунок після запуску

Для того, щоб активувати весь функціонал користувач має завантажити зображення на форму, скориставшись кнопкою «Завантажити». Після натискання на кнопку, з'явиться вікно для вибору зображення, яке користувачу треба обробити, показане на рисунку 3.4.

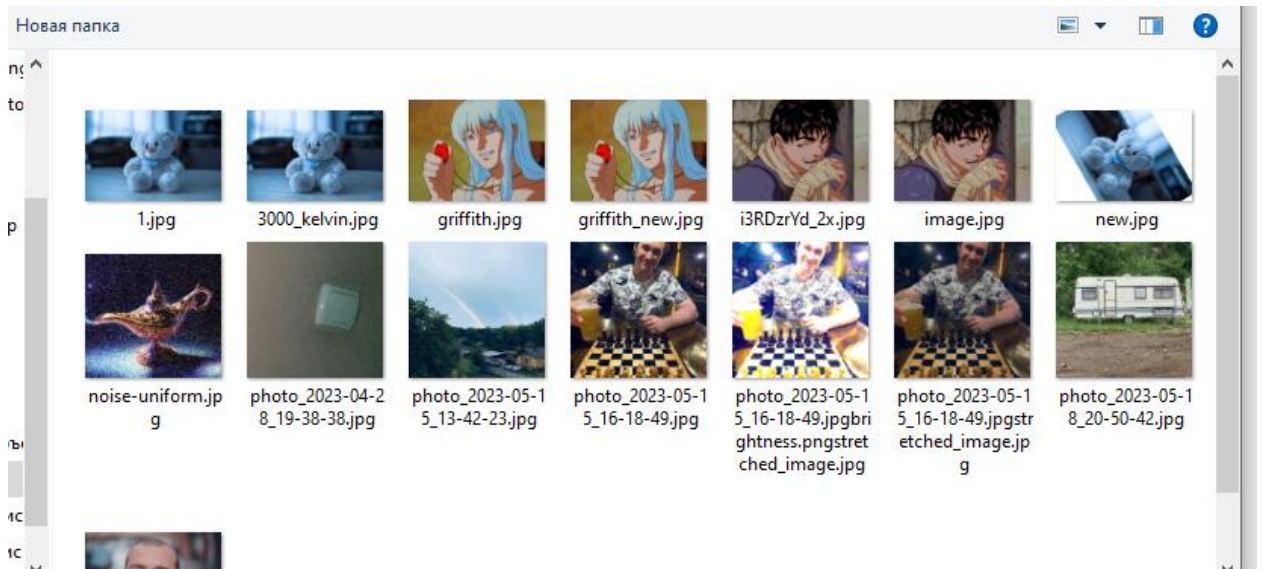


Рисунок 3.4 – Вибір необхідного зображення для обробки

Після вибору необхідного зображення користувач повертається на форму і бачить те, що зображення тепер відображається на формі і те, що усі кнопки тепер активні (рис. 3.5).

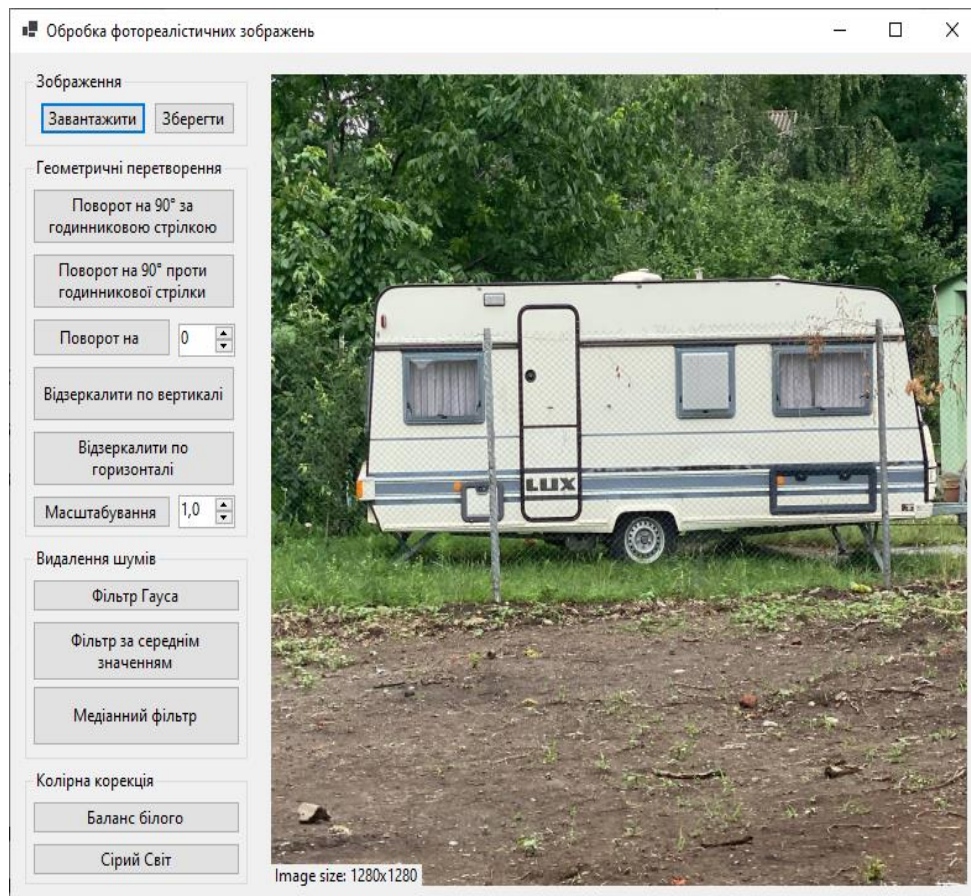


Рисунок 3.5 – Вигляд програми після вибору зображення

Після того, як увесь функціонал застосунку стає доступним користувач виконує усі необхідні йому методи обробки.

Обрізка зображення доступна, якщо користувач натисне і буде тримати ліву кнопку миші на необхідній зоні зображення, яку треба залишити (рис. 3.6).

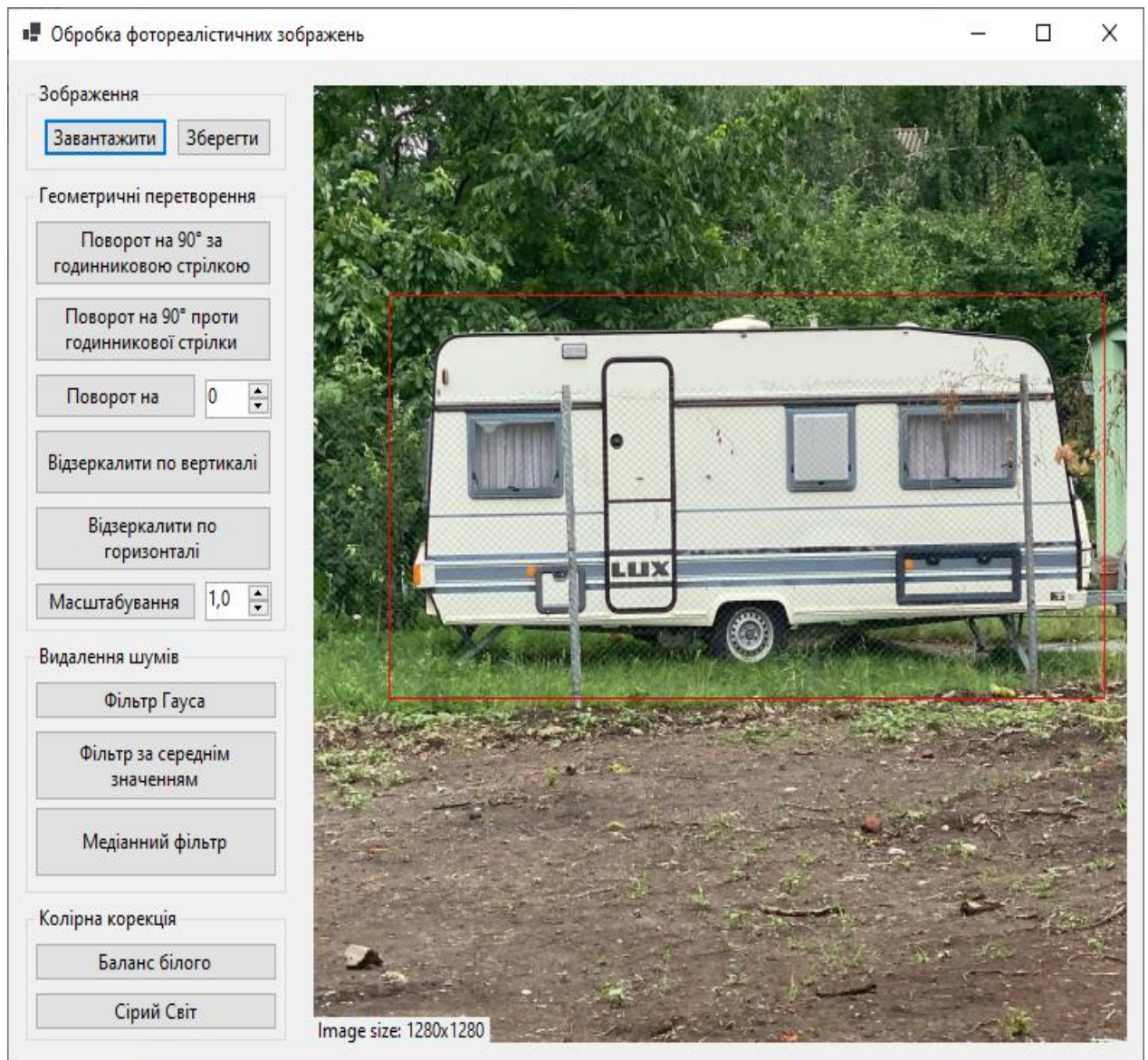


Рисунок 3.6 – Червона зона показує межі нового зображення

Після того, як користувач відпустить ліву кнопку миші, програма автоматично обріже зображення по обраним контурам. Після чого нове зображення одразу з'явиться на екрані, що продемонстровано на рисунку 3.7, після чого користувач може продовжити його обробляти.

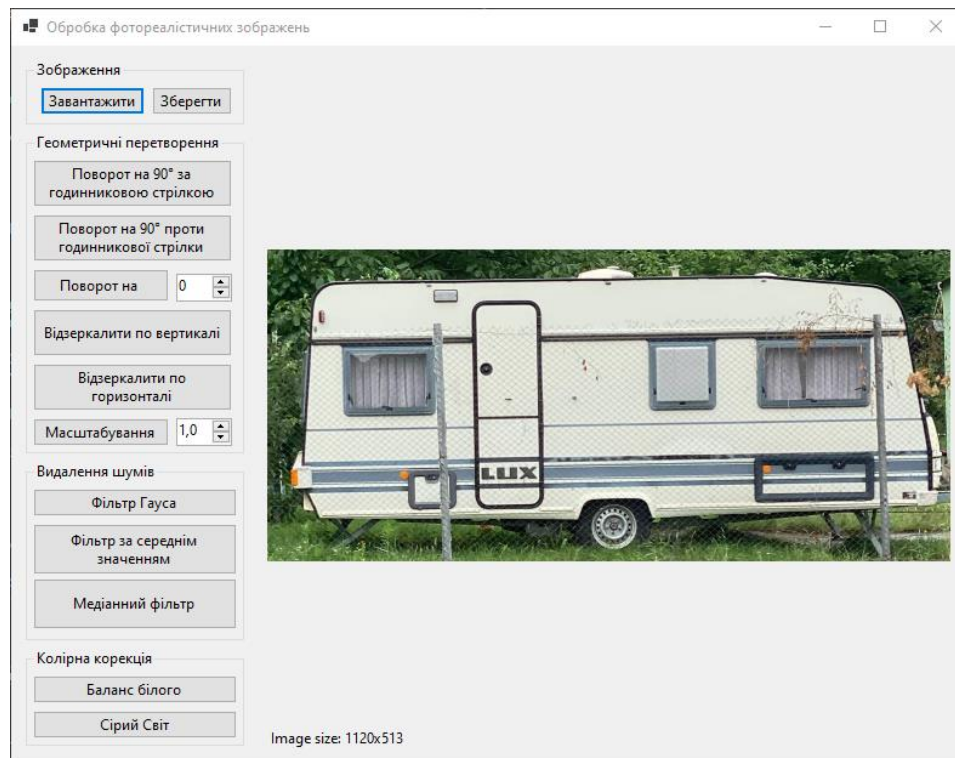


Рисунок 3.7 – Обрізане зображення

Після виконання усіх необхідних методів обробки користувач може зберегти нове зображення. Для цього необхідно натиснути кнопку «Зберегти», після чого з'явиться вікно (рис.3.8), в якому можна обрати куди зберегти зображення.

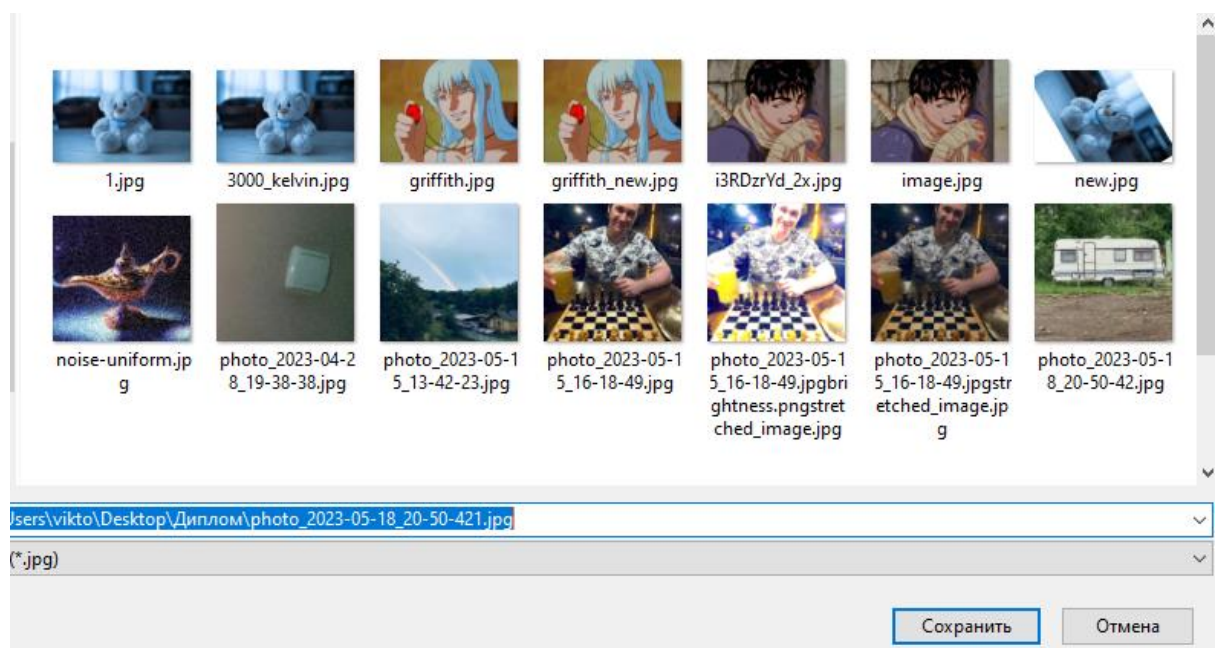


Рисунок 3.8 – Діалогове вікно для збереження обробленого зображення

### 3.4 Тестування розробленої моделі

Тестування застосунку проводиться з метою виявлення помилок, дефектів і недоліків, які можуть вплинути на функціональність та надійність програмного забезпечення. Основна мета тестування полягає в перевірці, чи працює застосунок так, як він повинен працювати, чи відповідає вимогам і очікуванням користувачів.

#### 3.4.1 Причини проведення тестування

Основні причини проведення тестування застосунку:

- виявлення помилок. Тестування допомагає виявити програмні помилки, дефекти та недоліки в застосунку. Це дозволяє розробникам виправити проблеми перед випуском продукту на ринок;
- перевірка функціональності. Тестування включає в себе перевірку відповідності функцій і можливостей застосунку вимогам, специфікаціям та очікуванням користувачів. Це допомагає забезпечити, що застосунок працює так, як очікується;
- перевірка взаємодії. Тестування також дозволяє перевірити взаємодію застосунку з іншими системами, платформами або пристроями. Наприклад, якщо застосунок взаємодіє з сервером, базою даних або зовнішніми сервісами, тестування допоможе переконатися, що ця взаємодія працює належним чином;
- надійність і стабільність. Тестування допомагає перевірити, наскільки стабільний та надійний застосунок. Це включає в себе перевірку його реакції на непередбачувані ситуації, обробку помилок і відновлення після збоїв;
- оптимізація продукту. Тестування допомагає виявити можливості для вдосконалення продукту. Під час тестування можна виявити шляхи

оптимізації, покращення швидкості, ефективності або інтерфейсу користувача.

Враховуючи ці причини, тестування застосунку є важливою складовою процесу розробки програмного забезпечення, оскільки воно допомагає забезпечити якість та задоволення користувачів.

### 3.4.2 Основні етапи тестування

Початок тестування застосунку вимагає планування та підготовки. Почати треба з уважного вивчення вимог до застосунку. Це допоможе зрозуміти, як повинен працювати застосунок та які очікування мають користувачі. Вимоги можуть бути визначені в специфікації, документації або у вигляді задач.

Після вивчення вимог необхідно створити план тестування, який охоплює основні аспекти, функціональності та сценарії, які потрібно перевірити. Необхідно обрати методи тестування, які відповідають потребам застосунку. Після чого треба створити детальні тестові сценарії, які включають набір кроків для перевірки різних функцій та можливостей застосунку. Це допоможе систематично тестувати різні аспекти застосунку і переконатися, що нічого не пропущено.

Потім необхідно підготувати тестове середовище, включаючи необхідне обладнання, програмне забезпечення та тестові дані. Треба впевнитися, що достатньо ресурсів для виконання тестів. Після чого можна починати запускати тестові сценарії, виконуючи кожен крок і перевіряючи результати. Необхідно записувати помилки, дефекти або неполадки, які зустрілися під час тестування.

В кінці треба оцінити результати тестування, проаналізувати виявлені проблеми та помилки і сформулювати звіт про тестування, який містить інформацію про виявлені дефекти, їх пріоритети та рекомендації щодо

виправлення. Після виправлення виявлених проблем повторно виконати тестування, щоб переконатися, що виправлення були ефективними та не спричинили нові дефекти.

Процес тестування може варіюватися в залежності від специфіки проєкту та методології розробки. Важливо також враховувати тестування на ранніх етапах розробки та використовувати автоматизоване тестування, де це можливо, для забезпечення більшої ефективності та повторюваності тестів.

### 3.4.3 Тестування застосунку

Тестування розробленого застосунку слід почати від самого запуску. Після запуску з'являється вікно зображене на рисунку 3.9.

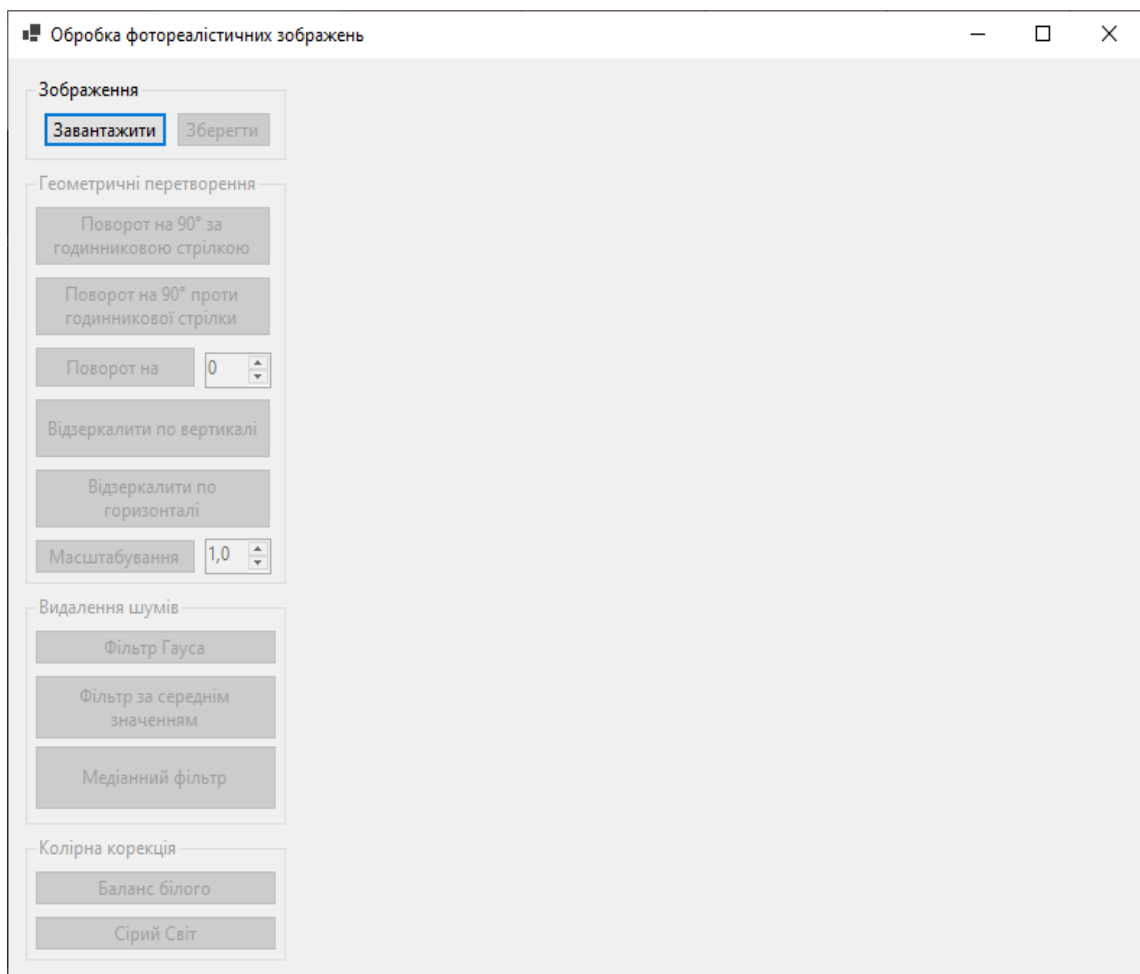


Рисунок 3.9 – Вікно застосунку після запуску

Після натискання на кнопку «Завантажити» з’являється діалогове вікно (рис. 3.10) з вибором зображень різного розширення, таких як JPG, PNG і BMP.

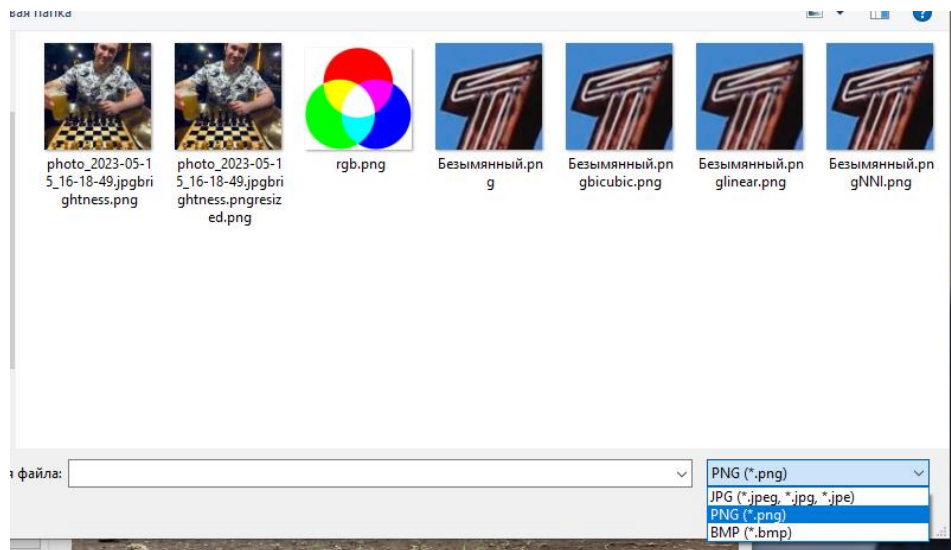


Рисунок 3.10 – Діалогове вікно з можливістю вибору формату зображення

Обравши зображення, воно з’являється на формі (рис. 3.11). Також, на рисунку 3.11 можна помітити, що в лівому нижньому куту зображення написано його розмір.

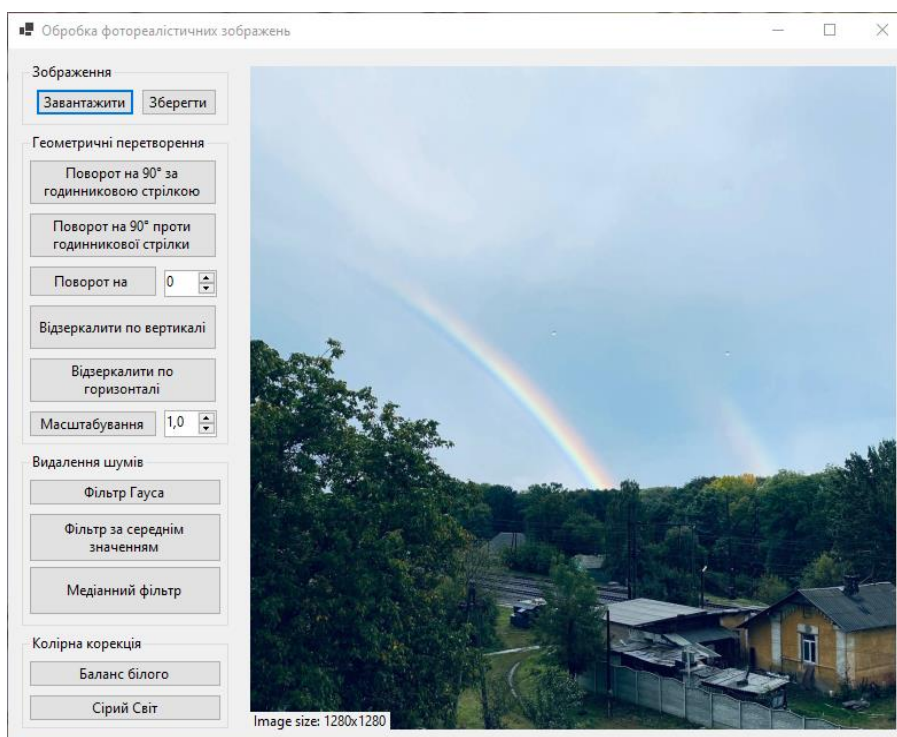


Рисунок 3.11 – Зображення користувача відображається

До цього моменту, під час тестування не було ніяких проблем з функціональністю застосунку. Тепер необхідно провести тестування саме алгоритмічної частини застосунку.

Після натискання на кнопку «Поворот на 90° за годинниковою стрілкою» зображення повернеться на 90 градусів за годинниковою стрілкою, що продемонстровано на рисунку 3.12.

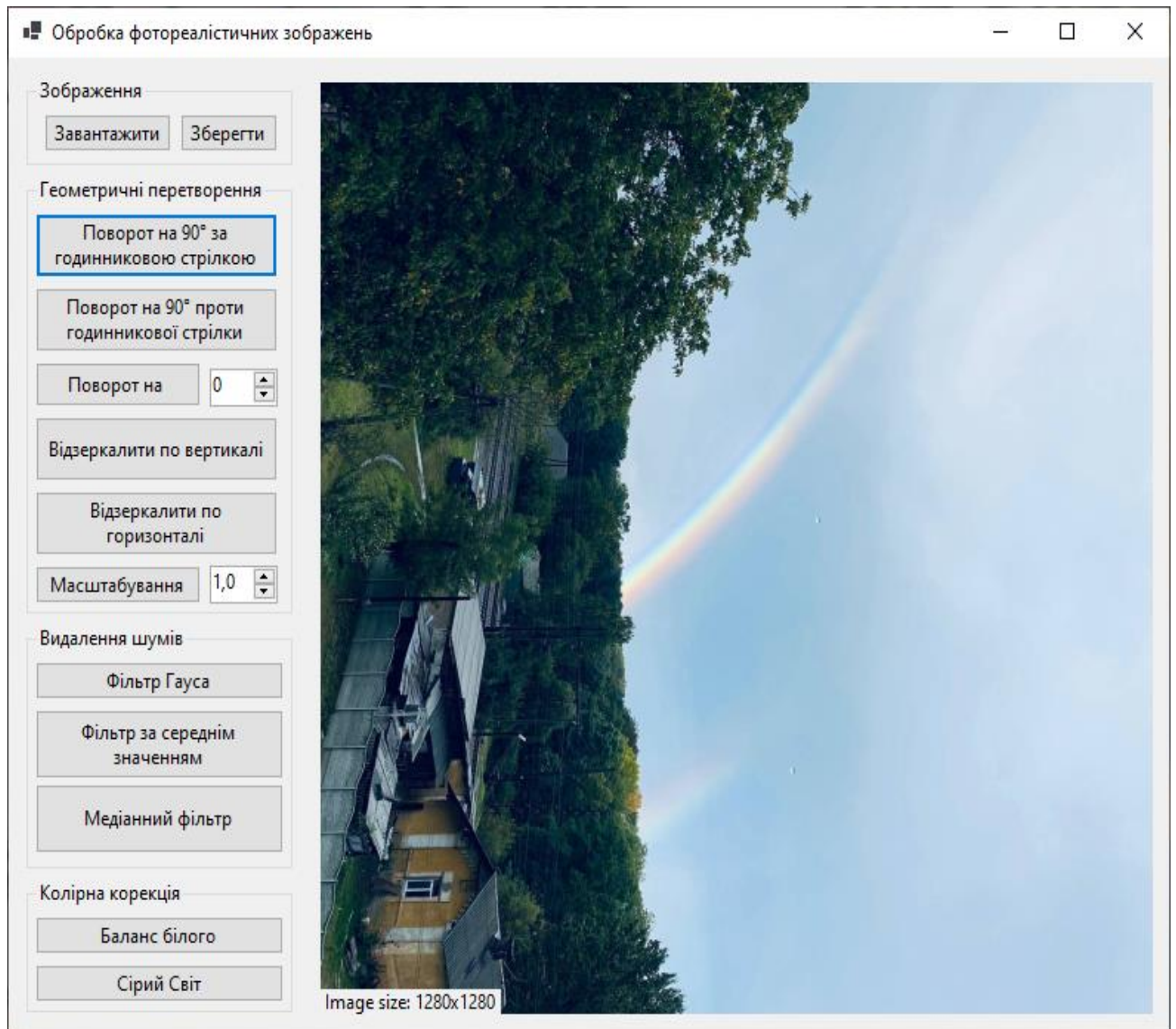


Рисунок 3.12 – Зображення після повороту на 90 градусів за годинниковою стрілкою

«Поворот на 90° проти годинникової стрілки» повертає зображення на 90 градусів проти годинникової стрілки, що показано на рисунку 3.13.

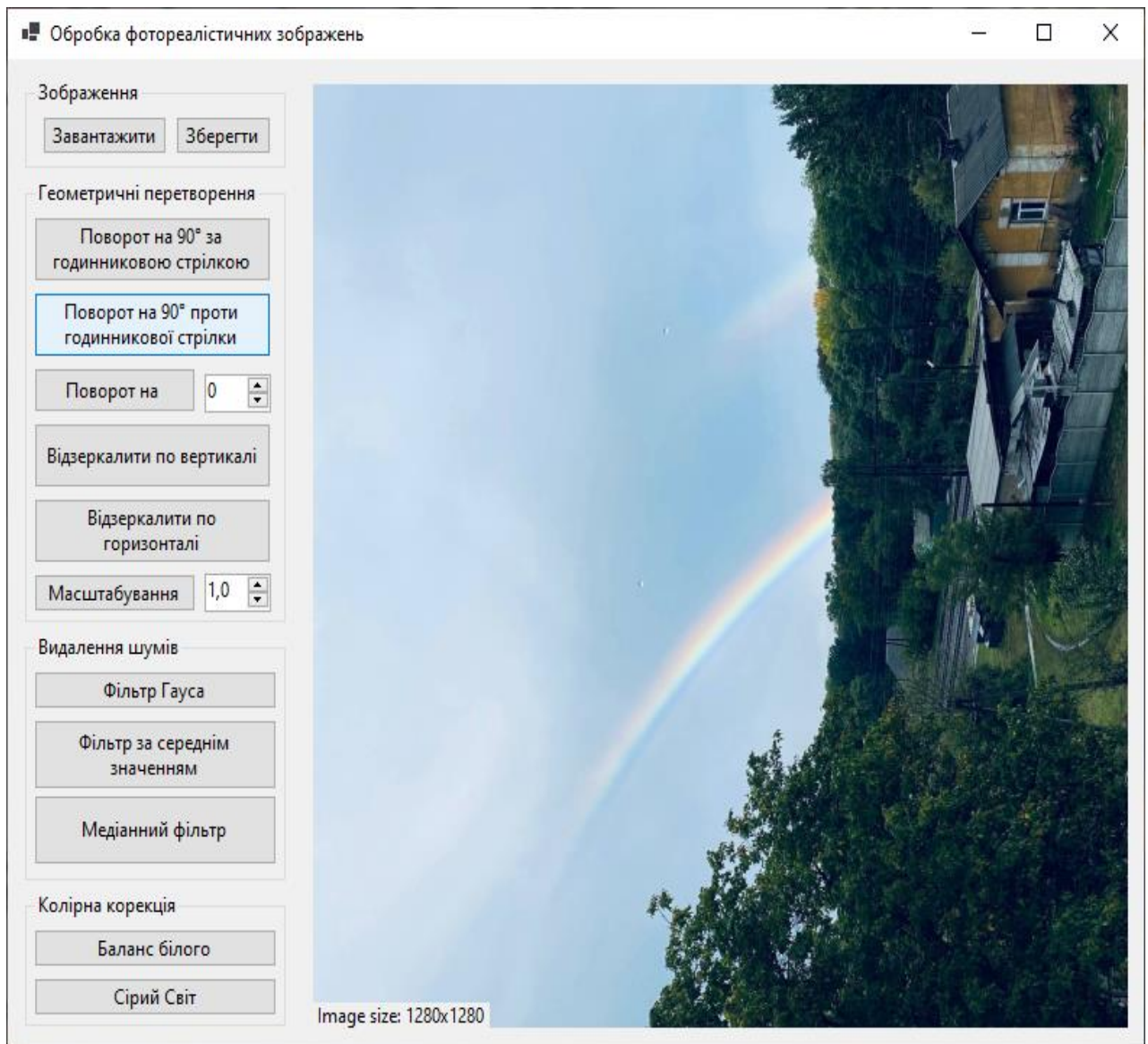


Рисунок 3.13 – Зображення після повороту на 90 градусів проти годинникової стрілки

Можна дійти висновку, що алгоритми на обертання на кратні градуси працюють як це необхідно користувачу.

Тепер необхідно перевірити обертання на кут який не є кратним до 90. Для цього необхідно встати курсором на поле для вводу і ввести с клавіатури значення, на яке необхідно повернути зображення або ж можна натискати стрілочки біля поля для вводу.

Нехай користувач задає значення куту повороту 70 градусів. В результаті буде отримано зображення з білими ділянками, що є небажаним ефектом, показано на рисунку 3.14.

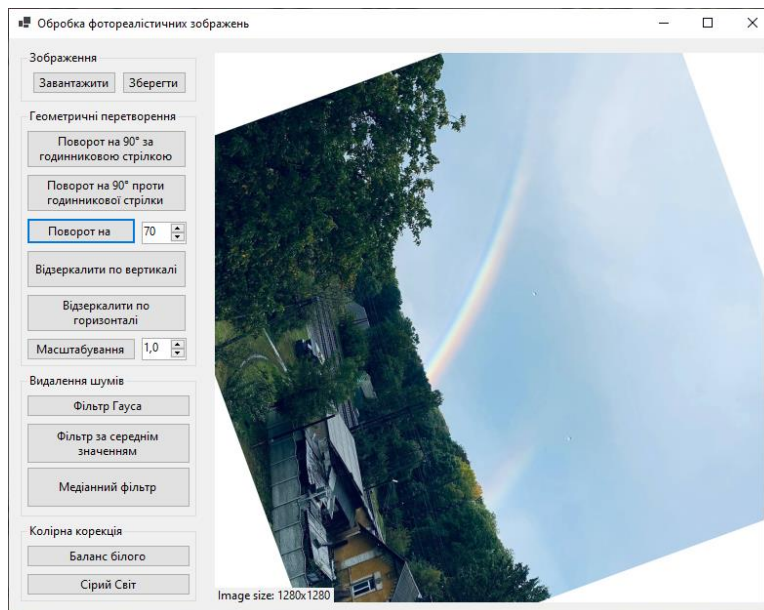


Рисунок 3.14 – Зображення після повороту на 70 градусів з обрізаними кутами

Справа в тому, що в методі повороту зображення останній параметр вказує на те, чи будуть обрізатися кути зображення після повороту чи ні. Необхідно змінити цей параметр і перевірити результат. На рисунку 3.15 зображена та ж сама операція, що й раніше, але тепер кути зображення не обрізані.

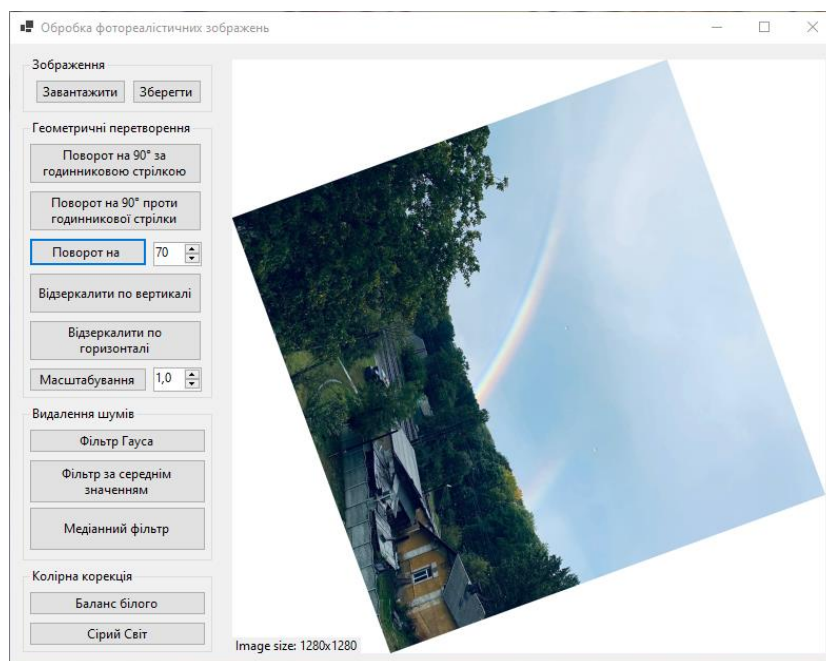


Рисунок 3.15 – Зображення після повороту на 70 градусів

Після виправлення недоліку з поворотом зображення, необхідно протестувати методи віддзеркалення зображення. Віддзеркалення зображення можна використовувати для створення цікавих візуальних ефектів або симетрії.

Після завантаження файлу, користувач може натиснути на кнопку «Віддзеркалити по вертикалі». Результат дії вертикального віддзеркалення показано на рисунку 3.16.

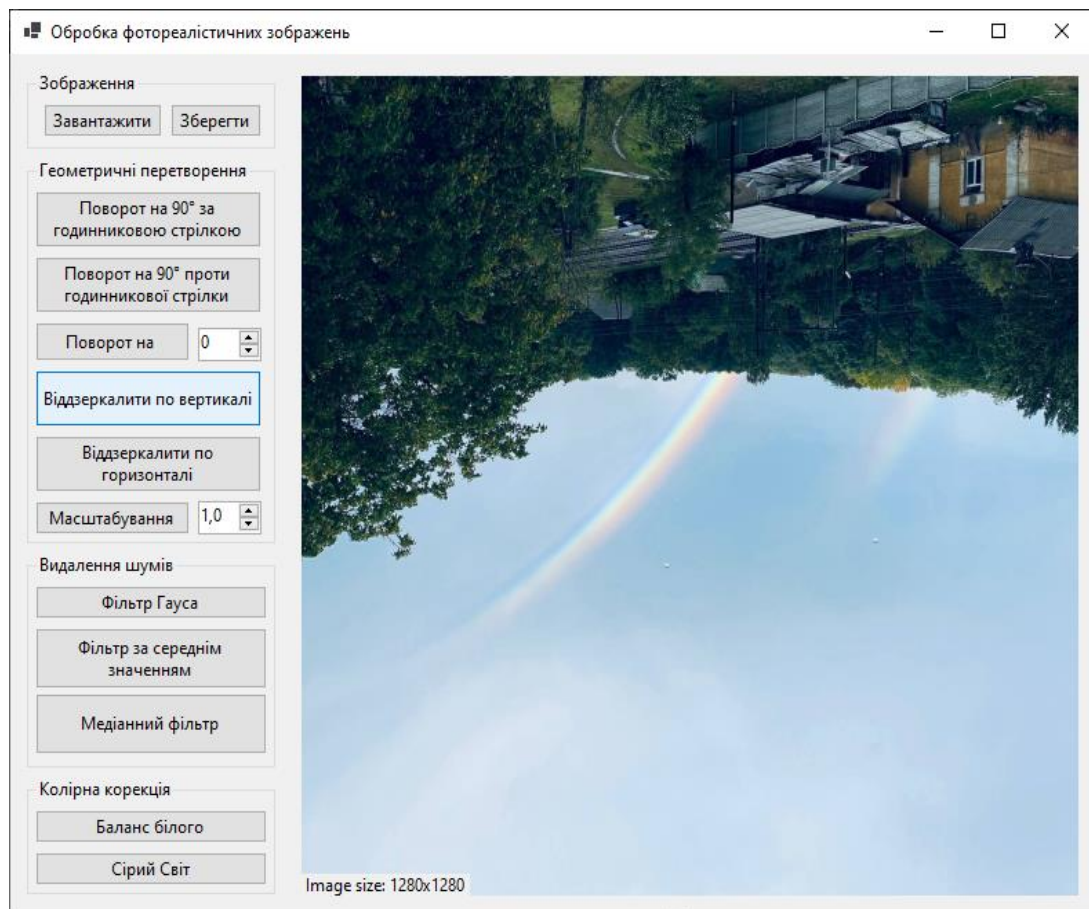


Рисунок 3.16 – Зображення після віддзеркалення по вертикалі

Також користувач може натиснути на кнопку «Віддзеркалити по горизонталі». Після віддзеркалення по горизонталі зображення набуває більш естетичного вигляду, що продемонстровано на рисунку 3.17.

Тепер необхідно перевірити як працює масштабування зображення. Збільшимо зображення в півтори рази. На рисунку 3.18 показано результат дії масштабування.

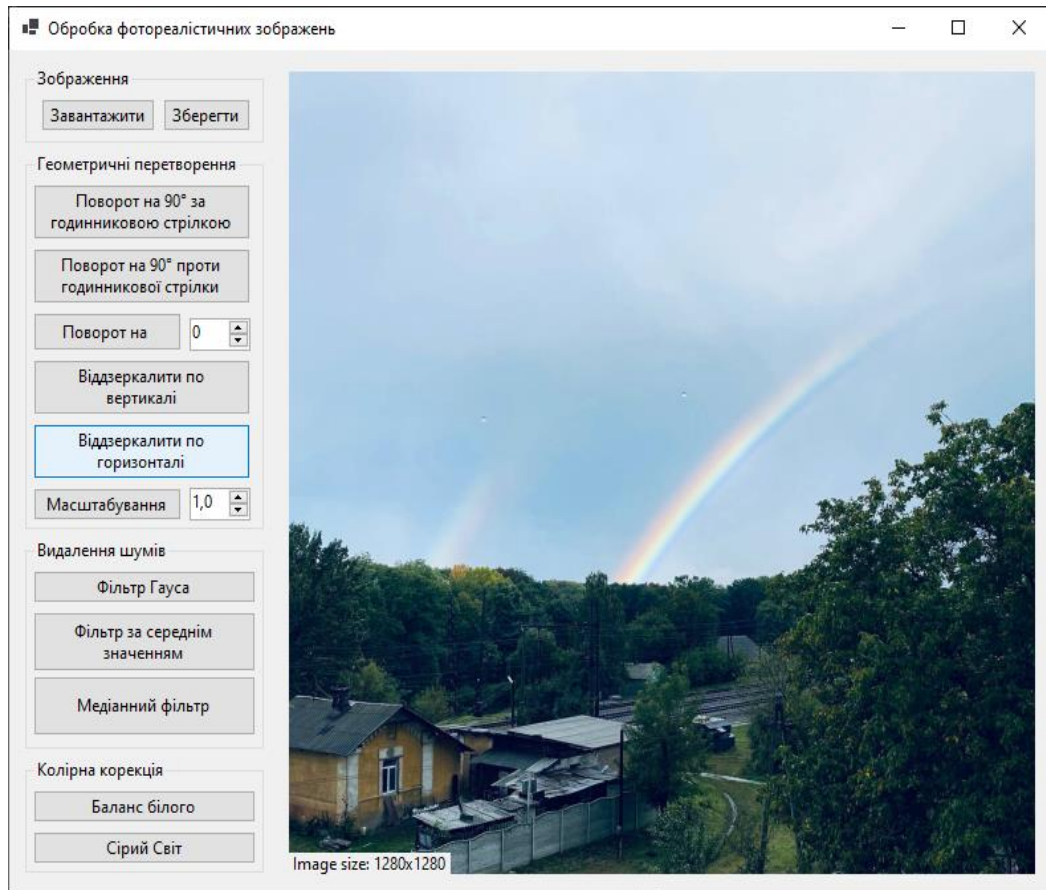


Рисунок 3.17 – Зображення після віддзеркалення по вертикалі

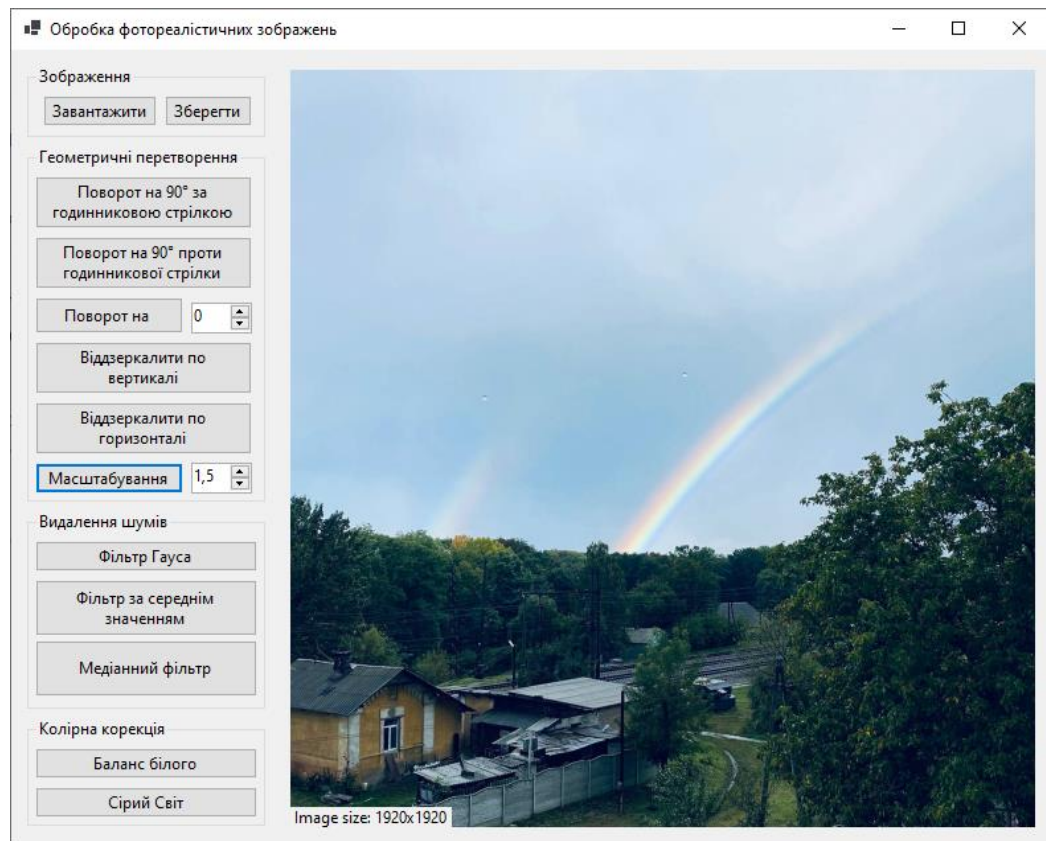


Рисунок 3.18 – Зображення після збільшення масштабу

Як можна помітити, зображення не змінює свій розмір відносно вікна, але в лівому нижньому куті можна помітити, що зображення замість  $1280 \times 1280$  пікселів має розміри  $1920 \times 1920$ .

Спробуємо зменшити зображення і перевірити чи працює масштаб з коефіцієнтом менше 1. Зменшення зображення продемонстровано на рисунку 3.19.

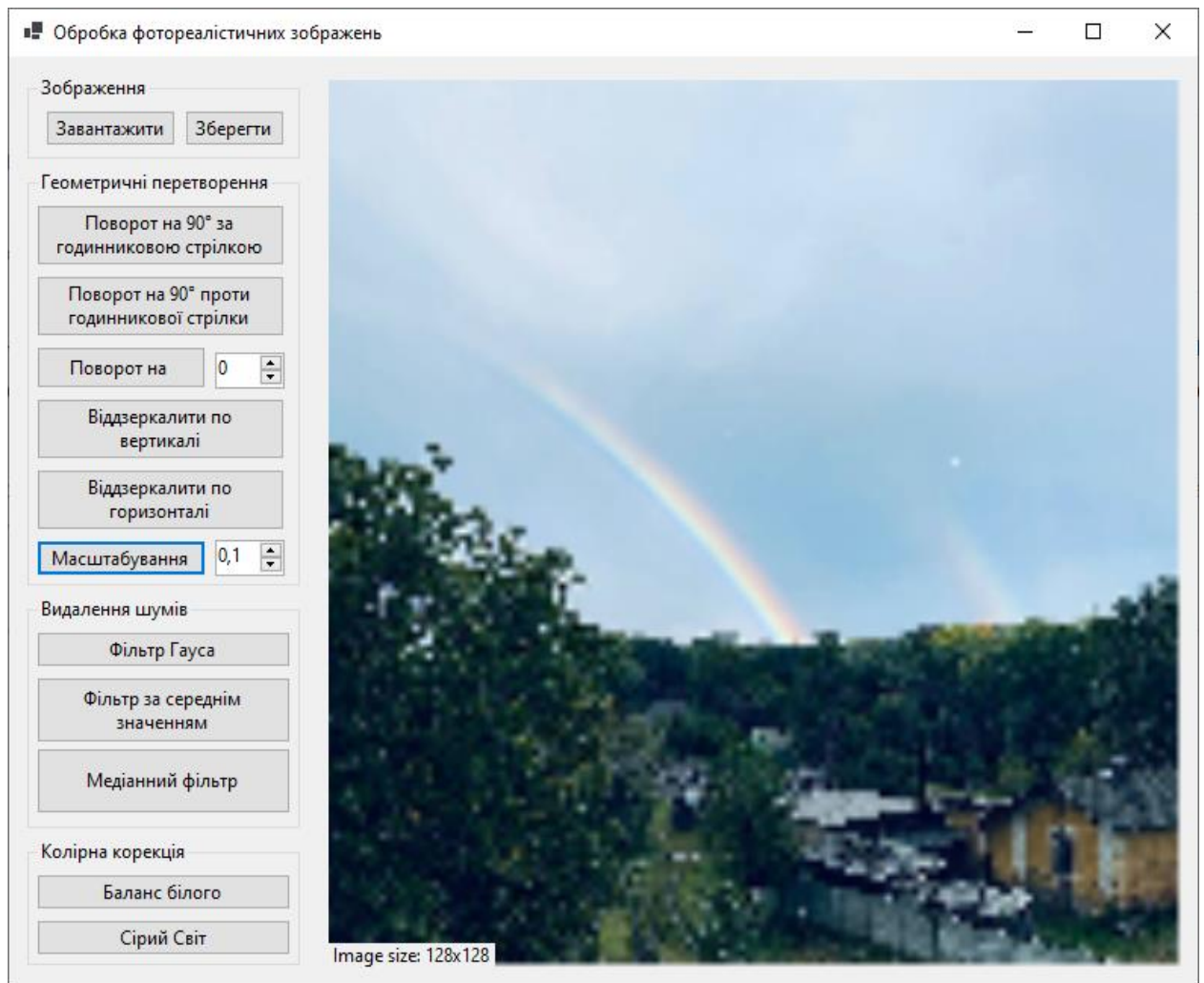


Рисунок 3.19 – Зображення після зменшення масштабу

Тепер треба перевірити що відбудеться, якщо користувач введе з клавіатури, наприклад, від'ємне значення масштабу (рис. 3.20).

Після введення від'ємного числа і натискання на кнопку масштабування відбувається так, що від'ємне значення в полі змінюється на мінімальне допустиме значення масштабування, тобто 0,1, що показано на рисунку 3.21.

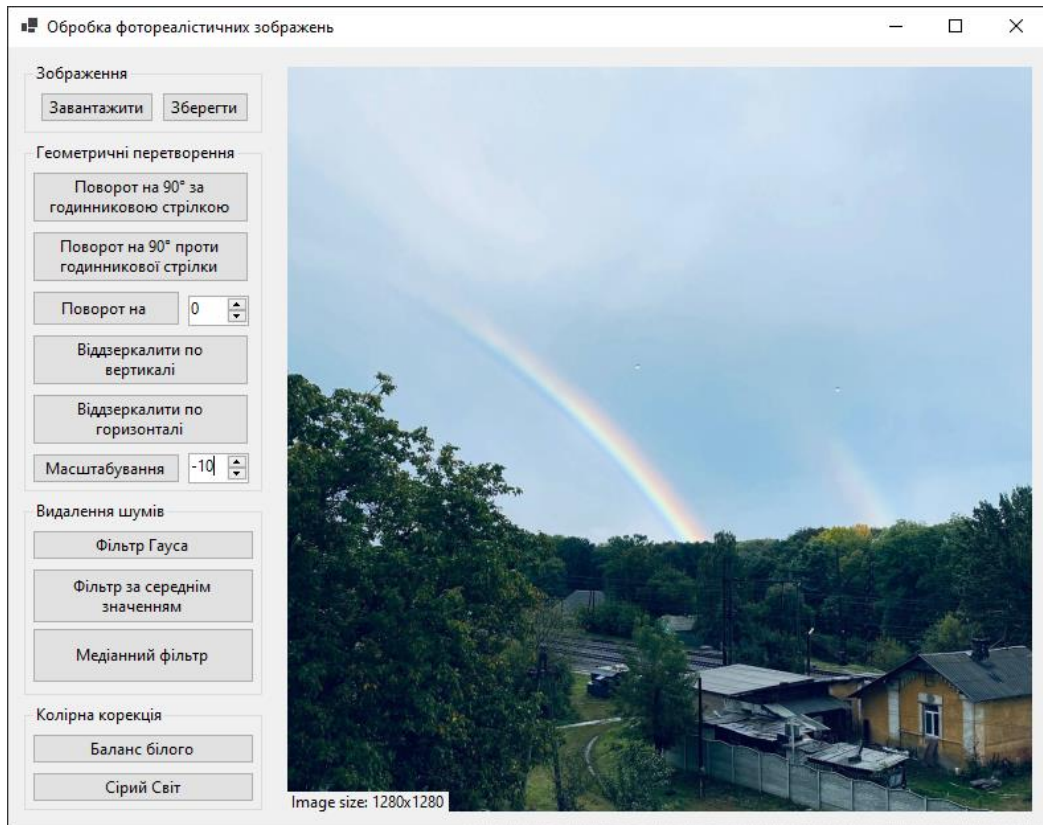


Рисунок 3.20 – Користувач намагається вписати в поле коефіцієнта масштабування від’ємне число

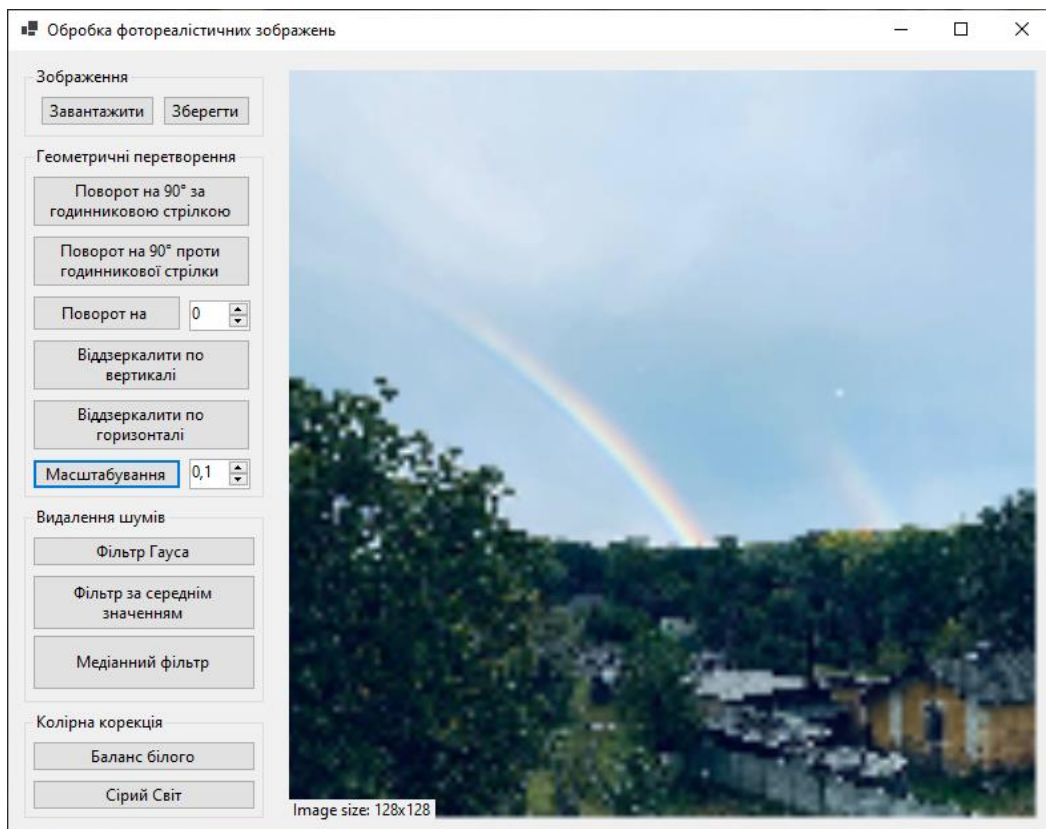


Рисунок 3.21 – Захист поля від від’ємних коефіцієнтів масштабування

Тепер необхідно протестувати роботи фільтрів для видалення шуму. Почати можна з фільтра Гауса. Користувачу необхідно завантажити зображення з шумом і після цього натиснути на кнопку «Фільтр Гауса». Після чого буде отримано розмите зображення з меншою кількістю шуму (рис. 3.22).

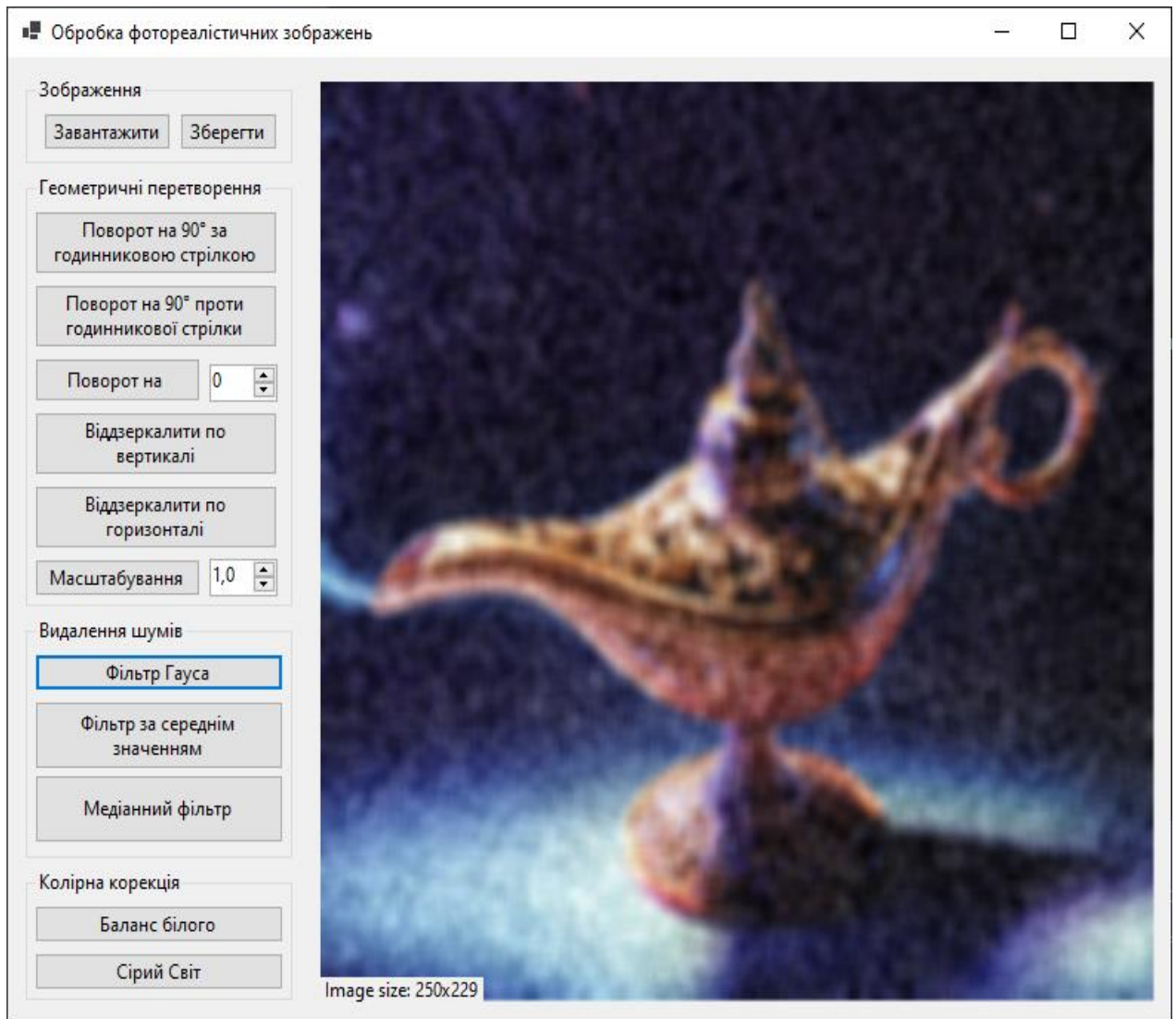


Рисунок 3.22 – Результат роботи фільтра Гауса

Можна побачити, що фільтр Гауса працює як необхідно користувачу. Однак сам фільтр Гауса не усуває шум безпосередньо, а лише згладжує його, роблячи його менш помітним. Інтенсивність шуму після застосування фільтра Гауса залежить від параметрів фільтра, таких як розмір ядра та стандартне відхилення. Тобто недоліком програми є те, що користувач не може налаштувати, наприклад, середньо-квадратичне відхилення.

Окрім фільтра Гауса, застосунок містить ще фільтр за середнім значенням і медіанний фільтри.

Фільтр за середнім значенням, як і фільтр Гауса не видаляє безпосередньо шум, а лише згладжує його. Результат роботи фільтра за середнім значенням показано на рисунку 3.23.

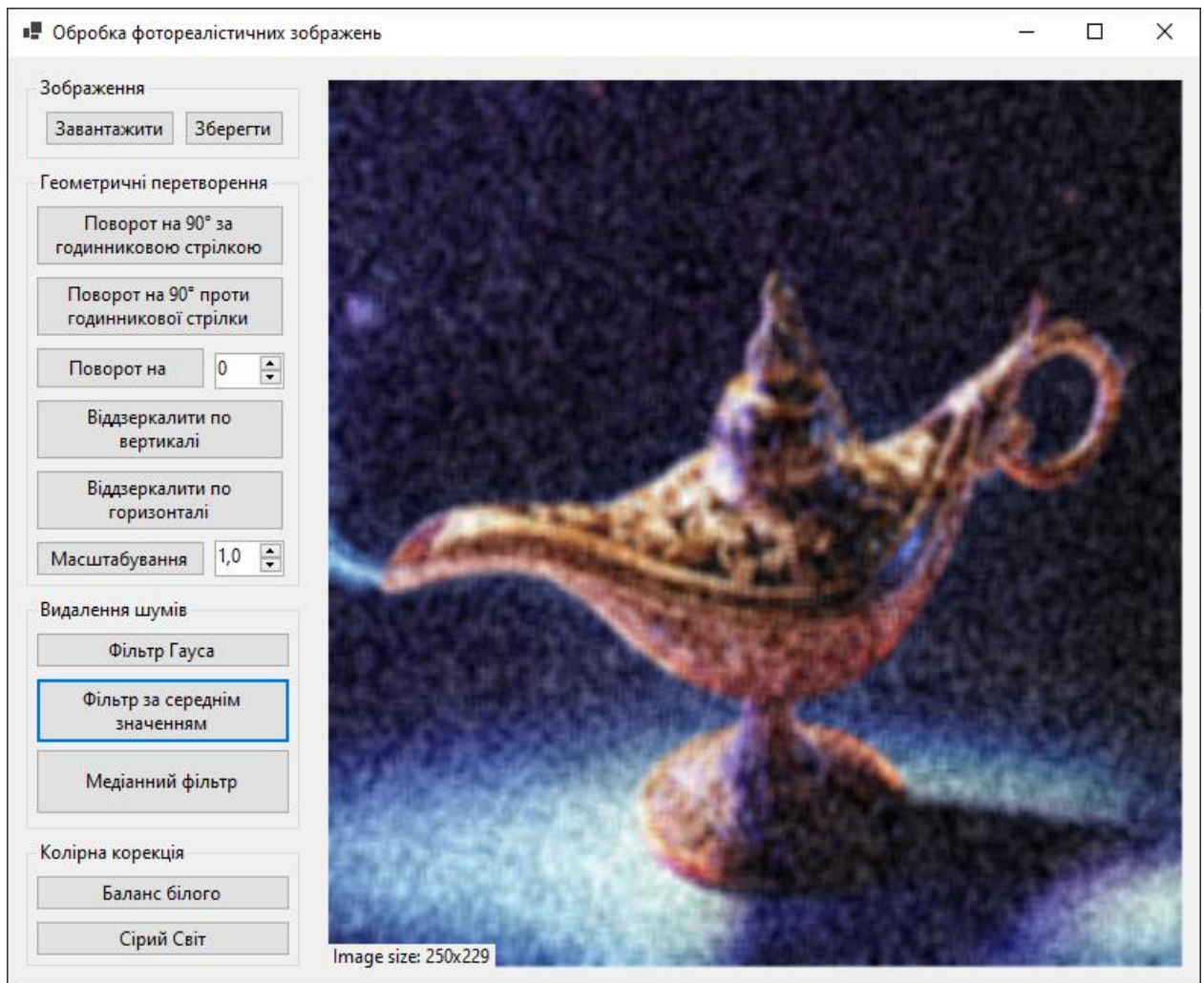


Рисунок 3.23 – Результат обробки фото за допомогою фільтра за середнім значенням

А медіанний фільтр в свою чергу видаляє шум з зображення, але недоліком є те, що видаляє лише «шум солі та перцю». Користувач має завантажити в застосунок зображення з ефектом «шуму солі та перцю» і натиснути на кнопку «Медіанний фільтр». Результат роботи медіанного фільтра показано на рисунку 3.24.

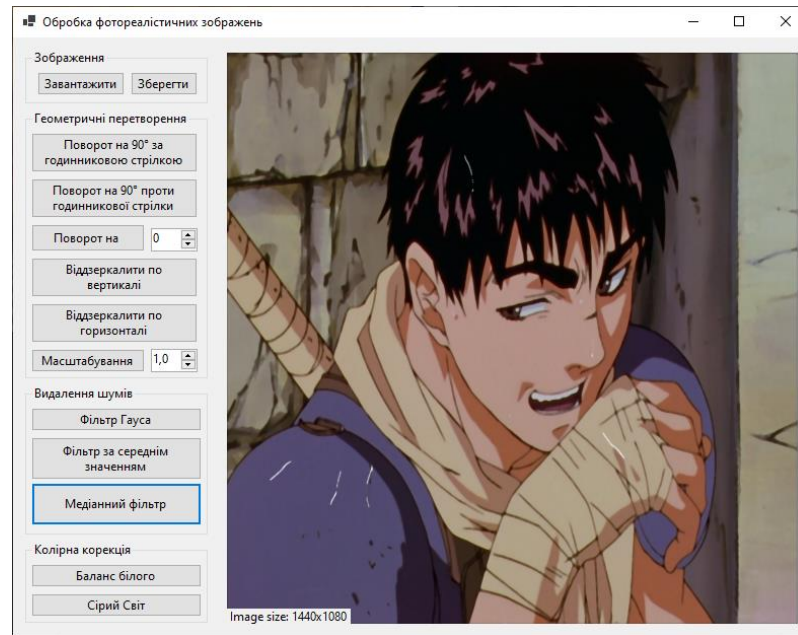


Рисунок 3.24 – Результат обробки фото за допомогою медіанного фільтра

Після перевірки фільтрів для видалення шуму, необхідно перевірити алгоритми колірної корекції. Для цього необхідно завантажити зображення з «холодним» або «теплим» освітленням для перевірки алгоритму сірого світу. Після цього треба натиснути кнопку «Сірий світ» і отримати результат, який продемонстровано на рисунку 3.25.

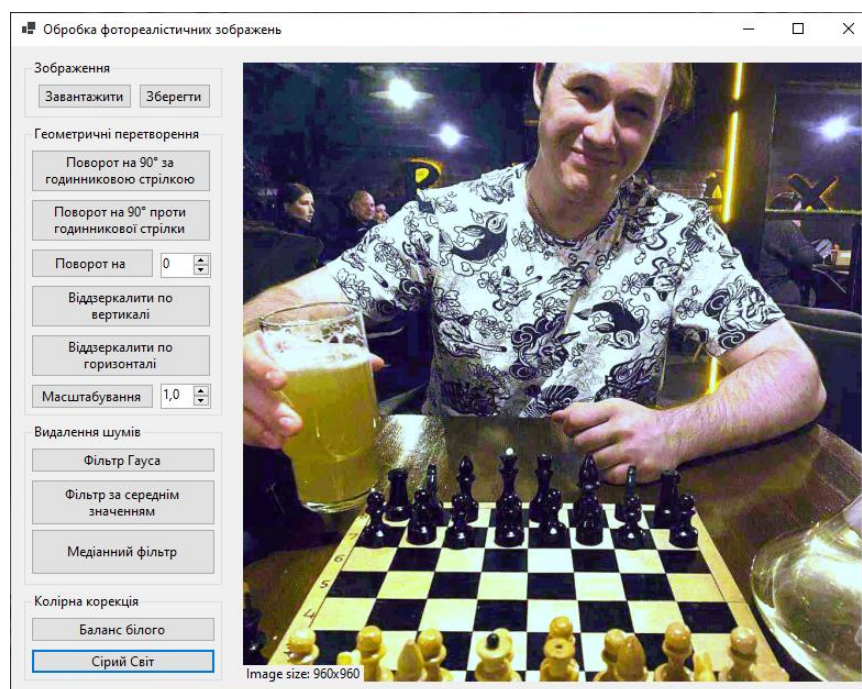


Рисунок 3.25 – Результат обробки фото алгоритмом сірого світу

Для перевірки алгоритму балансу білого необхідно завантажити зображення з поганим освітленням. Після цього треба натиснути кнопку «Баланс білого» і отримати результат, який продемонстровано на рисунку 3.26.

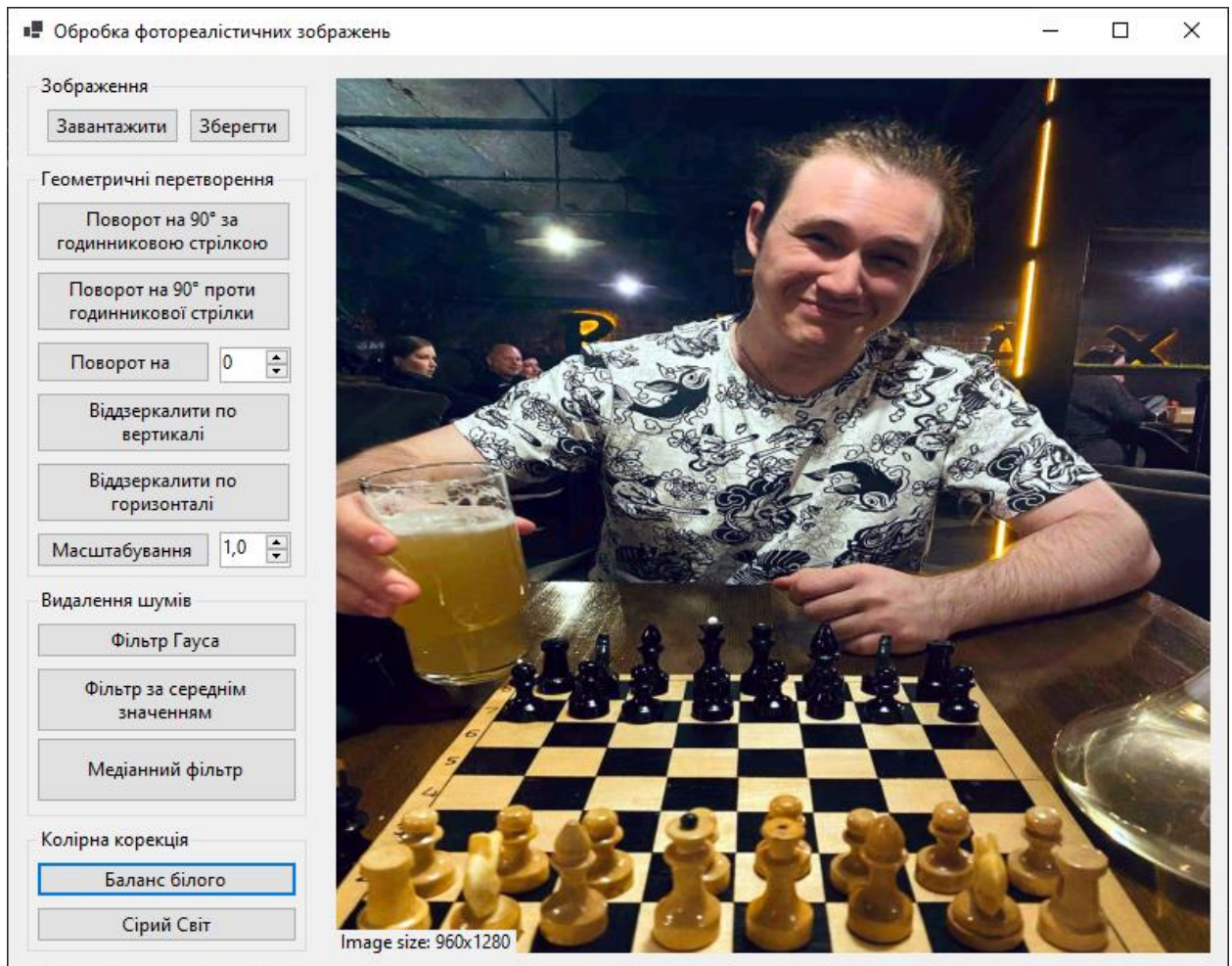


Рисунок 3.26 – Результат обробки фото алгоритмом балансу білого

Може статися таке, що користувач захоче забажає застосувати декілька алгоритмів обробки одночасно. Треба перевірити, що може статися в такому випадку. Нехай користувач вирішив спочатку застосувати баланс білого, після цього обрізати фото, збільшити зображення в 2 рази, повернувши його 90 градусів проти годинникової стрілки і віддзеркаливши горизонтально. Після виконання усіх цих дій, користувач зберігає зображення на комп'ютер. В результаті користувач отримує зображення показане на рисунку 3.27.



Рисунок 3.27 – Збережене на комп'ютер зображення після виконання декількох алгоритмів обробки одночасно

Тобто, програма передбачує як виконання усіх методів обробки зображення, як окремо, так і в об'єднанні з іншими. Недоліків у роботі застосунку не виявлено.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований застосунок для обробки фотореалістичних зображень за допомогою сучасних алгоритмів та методів обробки зображень.

Застосунок володіє потужним функціоналом, що дозволяє користувачеві здійснювати різноманітні операції з фотореалістичними зображеннями, такі як фільтрація, геометричні перетворення, регулювання контрасту і яскравості та інші.

Наукова новизна роботи полягає в застосуванні сучасних алгоритмів та методів обробки зображень для розробки застосунку, спроможного обробляти фотореалістичні зображення. Використання передових алгоритмів дозволило досягти високої якості обробки та точності результатів. Крім того, застосунок був проєктований з урахуванням зручності використання, надійності та швидкодії, що дозволяє користувачеві зручно та ефективно працювати з програмою.

Практична значущість роботи полягає в його гнучкості та можливостях для розширення функціоналу. Програмне розширення означає, що застосунок може бути легко модифікований або доповнений залежно від потреб користувача або змін у технологічному середовищі.

Отримані результати демонструють успішну реалізацію мети кваліфікаційної роботи – створення потужного безкоштовного інструменту для обробки фотореалістичних зображень. Застосунок може знайти своє застосування у різних галузях, де обробка зображень відіграє важливу роль.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. *Advances in Electrical and Electronic Engineering*, 21(1), 19-27.
2. Гороховатський, В. О., Передрій, О. О., Творошенко, І. С., & Марков, Т. Є. (2023). Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень.
3. Кобилін, О.А., & Творошенко, І.С. (2021). *Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.*
4. Загальні відомості про канали. URL: <https://helpx.adobe.com/ua/photoshop/using/channel-basics.html> (дата звернення 14.04.2023).
5. В.О. Гороховатський (2022). *Комплекс навчально-методичного забезпечення навчальної дисципліни «Обробка зображень та мультимедіа»:* навч. Посібник. *Харків: ХНУРЕ.*
6. HIAMS. URL: <http://ai-tern.in.ua/Filtration.html> (дата звернення 20.04.2023).
7. Офіційний Adobe Photoshop | Програма для редагування фото та розробки дизайну. URL: <https://www.adobe.com/ua/products/photoshop.html> (дата звернення 21.04.2023).
8. GIMP - GNU Image Manipulation Program. URL: <https://www.gimp.org/> (дата звернення 21.04.2023).
9. Google Photos. URL: [https://www.google.com/intl/en\\_uk/photos/about/](https://www.google.com/intl/en_uk/photos/about/) (дата звернення 21.04.2023).
10. Gorokhovatskyi, V., Tvoroshenko, I., & Chmutov, Y. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Advanced Information Systems*, 6(3), 5-12.

11. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for Fast Metric Data Search in Structural Methods for Image Classification. *IEEE Access*, 10, 124738-124746.

12. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

13. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2021). Methods of classification of images on the basis of the values of statistical distributions for the composition of structural description components. *IEEE Access*, 9, 92964-92973.

14. Spatial Filters - Gaussian Smoothing. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm> (дата звернення 24.04.2023).

15. Gaussian filter, or Gaussian blur — Librow — Digital LCD dashboards for cars and boats. URL: <http://www.librow.com/articles/article-9> (дата звернення 24.04.2023).

16. Гороховатський, В.О., & Творошенко, І.С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.

17. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). Image classification method modification based on model of logic processing of bit description weights vector. *Telecommunications and Radio Engineering*, 79(1).

18. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Cluster representation of the structural description of images for effective classification.

19. Gorokhovatskyi, V., Tvoroshenko, I., & Chmutov, Y. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Advanced Information Systems*, 6(3), 5-12.

20. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Al-Dhaifallah, M. (2022). Classification of images based on a system of hierarchical features. *Computers, Materials & Continua*, 72(1), 1785-1797.

21. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image classification based on the Kohonen network and the data space modification.

22. White Balancing — An Enhancement Technique in Image Processing | by Matt Maulion | Medium. URL: <https://mattmaulion.medium.com/white-balancing-an-enhancement-technique-in-image-processing-8dd773c69f6> (дата звернення 26.04.2023).

23. Visual Studio: IDE and Code Editor for Software Developers and Teams. URL: <https://visualstudio.microsoft.com/> (дата звернення 03.05.2023).

24. C# docs - get started, tutorials, reference. | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення 03.05.2023).

25. Windows GDI - Win32 apps | Microsoft Learn. URL: <https://learn.microsoft.com/uk-ua/windows/win32/gdi/windows-gdi> (дата звернення 04.05.2023).

26. What is Windows Forms - Windows Forms .NET | Microsoft Learn. URL: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/?view=netdesktop-7.0> (дата звернення 04.05.2023).

27. Windows Forms for .NET 7 documentation | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-6.0> (дата звернення 04.05.2023).

28. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

29. OpenCV documentation index. URL: <https://docs.opencv.org/> (дата звернення 07.05.2023).

30. Emgu CV Library Documentation. URL: [https://www.emgu.com/wiki/files/4.7.0/document/html/R\\_Project\\_Emgu\\_CV\\_Documentation.htm](https://www.emgu.com/wiki/files/4.7.0/document/html/R_Project_Emgu_CV_Documentation.htm) (дата звернення 07.05.2023).