

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНА РОБОТА

Метод та програмні засоби масштабування зображень

Назва теми

Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр ДППЗ.170115.01.08.ПЗ

Виконав студент 2 курсу група ПЗм-21-1



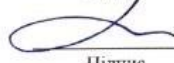
Підпис

В.А. Павлюк

Ініціали, прізвище

Керівник д-р фіз.-мат. наук, проф.

Науковий ступінь, звання



Підпис

Л. П. Бедратюк

Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент



Підпис

Г. І. Радельчук

Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення



Підпис

Л. П. Бедратюк


Ініціали, прізвище

2 грудня 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри 173
Л. П. Бедратюк 
01 09 2022 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Павлюку Віталію Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Метод та програмні засоби масштабування зображень

Керівник проєкту (роботи) Бедратюк Леонід Петрович, д-р фіз.-мат. наук, проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчєне звання

Затверджена наказом ректора університету від 01.07.2022 р. № 83

2. Строк подання студентом проєкту (роботи) на кафедру 01.12.2022 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1 Дослідження предметної області та постановка задачі

2 Концепції, моделі та методи вирішення задачі

3 Технології вирішення задачі

4 Реалізація та тестування програмного засобу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Гурман І. В., доцент	 23.11.2022	 1.12.2022
Нормоконтроль	Радельчук Г. І., доцент	 23.11.2022	 1.12.2022

7. Дата видачі завдання « 01 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Вивчення предметної області; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження;	01.09-10.09.2022	
2 Робота над розділом 1 дипломної роботи аналіз відомих моделей, методів та засобів за темою роботи; визначення методологічних підходів до вирішення задачі; висновки до розділу та постановка задач дослідження	11.09-25.09.2022	
3 Робота над розділом 2 дипломної роботи – розробка моделей, методів та алгоритмів вирішення задачі; висновки до розділу	26.09-10.10.2022	
4 Робота над науковими статтями	11.10-30.10.2022	
5 Робота над розділом 3 дипломної роботи – розробка інформаційної технології вирішення задачі (аналіз вимог до програмного засобу та його проектування, аналіз та вибір засобів реалізації програми); висновки до розділу	11.10-26.10.2022	
6 Робота над розділом 4 дипломної роботи – програмна реалізація спроектованих рішень, результати експериментів та їх аналіз; дослідження ефективності запропонованих рішень; висновки до розділу	27.10-17.11.2022	
7 Попередній захист дипломної роботи	Листопад	
8 Перевірка роботи на наявність плагіату; нормоконтроль; брошурування пояснювальної записки; підготовка супровідних документів	01.12-04.12.2022	
9 Підготовка до захисту дипломної роботи	з 01.12.2022 р.	

Студент


Підпис

В. А. Павлюк

Ініціали, прізвище

Керівник проєкту (роботи)


Підпис

Л. П. Бедратюк

Ініціали, прізвище

РЕФЕРАТ

Тема дипломної роботи: «Метод та програмні засоби масштабування зображень».

Автор роботи: Павлюк Віталій Андрійович.

Керівник роботи: Бедратюк Леонід Петрович.

Пояснювальна записка: 95 с., 33 рис., 3 табл., 3 дод., 18 джерел.

Графічна частина: 15 презентаційних слайдів.

МАСШТАБУВАННЯ, ОБРОБКА ЗОБРАЖЕННЯ, ЯДРА ЗОБРАЖЕННЯ, ШИРОКИЙ ДИНАМІЧНИЙ ДІАПАЗОН, ПРОГРАМНИЙ ЗАСІБ

Об'єкт дослідження – масштабування зображень.

Предмет дослідження – методи, алгоритми та програмні засоби масштабування растрових зображень.

Мета дослідження – розробка методу масштабування зображень, який не буде використовувати велику кількість ресурсів та буде поєднувати збільшення роздільної здатності зображення із покращенням його кольорового тону.

У роботі використані наступні методи дослідження та апаратура:

- спостереження, експеримент, абстрагування, аналіз та синтез, формалізація;
- інструментальні засоби проектування, програмування та тестування;
- персональний комп'ютер.

Під час дипломного проектування було досліджено галузь масштабування та покращення зображень, а також було визначено деякі невирішені проблеми у галузі та методологічні підходи до їх вирішення; на базі невирішених проблем розроблено власний метод забезпечення масштабування зображення, а також виконано його програмну реалізацію.

Розроблений у процесі дипломного проектування метод пропонує свій алгоритм масштабування та покращення растрових зображень, шляхом комбінації уже існуючих методів обробки зображення.

Для реалізації програмного засобу використано технології OpenCV, Pillow та інші, а для тестування готового програмного засобу – метрики схожості зображень.

Проведені емпіричні дослідження свідчать про адекватність та ефективність розробленого методу масштабування зображень, а також про його працездатність та функціональну придатність реалізованого на його основі програмного засобу.

Апробація отриманих результатів показала, що результати масштабування є схожими із оригінальними зображеннями, що свідчить про відсутність їх деформацій та трансформування, а також показала оптимізованим по різних сценаріях виконання. Отже, розроблений метод можна рекомендувати до використання використання компаніям, які займаються обробкою зображень та графічним дизайнерам.



Підпис

01.12.2022
Дата

ABSTRACT

Master's thesis: «Image scaling method and software».

Author: Pavliuk Vitalii.

Head of research: Bedratiuk Leonid.

Master's thesis consists of: 95 p., 33 pc., 3 tb., 3 add., 18 srs.

SCALING, IMAGE PROCESSING, IMAGE KERNELS, HIGH DYNAMIC RANGE, SOFTWARE

The object of research is image scaling.

The subject of research is methods, algorithms and software tools for scaling raster images.

The aim of the research is to develop a method of image scaling that will not use a large amount of resources and will combine an increase in image resolution with an improvement in its color tone.

The following approaches of the research are used during the study:

- monitoring, experiment, abstraction, analysis and synthesis, formalization;
- modern instrumental approaches of system design and development;
- personal computer.

During the study the field of image scaling and enhancement was investigated, and some unsolved problems in the field and methodological approaches to their solution were identified; on the basis of unsolved problems, an own method of ensuring image scaling was developed, and its software implementation was also performed.

The method developed in the process of diploma design offers its own algorithm for scaling and improving raster images by combining already existing image processing methods.

OpenCV, Pillow and other technologies were used to implement the software tool, and image similarity metrics were used to test the finished software tool.

The conducted empirical studies testify to the adequacy and effectiveness of the developed image scaling method, as well as to its workability and functional suitability of the software implemented on its basis.

Approbation of the obtained results showed that the scaling results are similar to the original images, which indicates the absence of their deformations and transformations, and also showed the optimization of different performance scenarios. Therefore, the developed method can be recommended for use by image processing companies and graphic designers.



Signature

01.12.2022

Date

ЗМІСТ

Перелік скорочень	9
Вступ.....	10
1 Дослідження предметної області та постановка задачі.....	14
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	14
1.2 Аналіз існуючих методів та програмних засобів масштабування зображення та їх змістовний опис.....	15
1.2.1 Real-ESRGAN	15
1.2.2 Gigapixel AI.....	17
1.2.3 LapSRN	19
1.2.4 Remini AI та GFPGAN	21
1.2.5 Adobe Photoshop	24
1.3 Методологічні підходи до вирішення задачі масштабування зображення.....	25
1.4 Висновки. Постановка задачі.....	27
2 Концепції, моделі та методи вирішення задачі.....	29
2.1 Концепції масштабування зображень	29
2.2 Моделі та методи масштабування зображення.....	36
2.3 Висновки	42
3 Технологія реалізації методу масштабування та покращення зображень	44
3.1 Аналіз вимог до програмного засобу	44
3.2 Проектування програмного засобу.....	46
3.3 Аналіз та вибір засобів програмної реалізації методу	50
3.4 Висновки	52
4 Реалізація та тестування програмної системи.....	54
4.1 Програмна реалізація	54
4.1.1 Структура та призначення модулів програми, їх взаємозв'язок.....	54
4.1.2 Розробка програмних компонентів	55
4.2 Результати тестування програмного засобу та їх аналіз.....	58

4.2.1 Вибір методу тестування.....	58
4.2 Розробка тестових сценаріїв	60
4.3 Аналіз результатів тестування.....	61
4.4 Інтеграція та налаштування програмного засобу	68
4.5 Висновки	69
Висновки	70
Додаток А Програмний код.....	75
Додаток Б Копія наукової публікації	80
Додаток В Презентаційні матеріали.....	87

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	–	програмний засіб
ПС	–	програмна система
GAN	–	Generative Adversarial Network
LR	–	Low-Resolution
SR	–	Super-Resolution
ESR	–	Real Enhanced Super-Resolution Generative Adversarial Networks
SRN	–	Super-Resolution network
AI	–	Artificial Intelligence
AHE	–	Adaptive Histogram Equalization
CLAHE	–	Contrast Limited Adaptive Histogram Equalization
CDF	–	Cumulative Distribution Function
HDR	–	High Dynamic Range
GUI	–	Graphical User Interface

ВСТУП

У сучасному світі цифрові зображення відіграють дуже важливу роль, їх легко зберігати, передавати та змінювати; вони широко використовуються в різних сферах. Оскільки вибірка цифрового зображення є непослідовною та неповною, роздільна здатність цифрового зображення часто обмежена. Тому під час детального перегляду частин зображення необхідно використовувати технології масштабування зображення. Якість зображення зазвичай визначається чинниками в природному середовищі. Фактори, присутні в природному середовищі, зазвичай пов'язані зі світлом. Якщо розподіл світла екстремальний, цільовий об'єкт на зображенні стає важко ідентифікувати. У цьому дослідженні пропонується метод масштабування зображення, який поєднує поліпшення зображення з його масштабування.

Поліпшення зображення в основному покращує сприйняття інформації в зображеннях для глядачів і забезпечує «кращий» вхід для інших автоматизованих методів обробки зображень. Основним мотивом покращення зображення є зміна атрибутів даного зображення, щоб зробити його придатним для певного завдання та спостерігача. Процес модифікації може відрізнитися залежно від поставленого завдання. Крім того, відповідно до вимог можна змінити більше одного атрибута зображення. Існують різні методи покращення зображення, і їх вибір може відрізнитися залежно від факторів, характерних для спостерігача, тобто зорова система людини та її досвід можуть додати значної частки суб'єктивності процедурі вибору. Для зорового сприйняття кольорові зображення дають більше інформації, ніж чорно-білі. Покращення кольору зображення відіграє важливу роль у цифровій обробці зображень [1].

Одними із перших, хто розглянув проблему масштабування зображення та їх низької роздільної здатності були Агелос Константінос Кацаггелос. Він описав у [2] процес відновлення цифрового зображення, а у роботі [3] розглянув проблему

перетворення зображень низької роздільної здатності (LR) у зображення надвисокої роздільної здатності (SR), а також у роботі [4] розглянув тему зменшення шуму для динамічних послідовностей зображень.

Завданням магістерської роботи є:

- дослідити основні проблеми існуючих методів масштабування та покращення зображення;

- визначити чіткі цілі для розробки методу;

- реалізація методу масштабування зображення.

Актуальність роботи полягає у тому, що дуже часто растрові зображення мають низьку роздільну здатність та низьку якість, у випадку чого виникає необхідність у їх масштабуванні із мінімальною кількістю втрат якості.

Об'єкт дослідження – масштабування зображень.

Предмет дослідження – методи, алгоритми та програмні засоби масштабування растрових зображень.

Мета дослідження – розробка методу масштабування зображення, який не буде використовувати велику кількість ресурсів та буде поєднувати збільшення роздільної здатності зображення із покращенням його кольорового тону.

Задля досягнення поставленої мети потрібно вирішити наступні задачі:

- проаналізувати та визначити специфіку предметної області;

- визначити основні переваги та недоліки існуючих методів масштабування цифрових зображень;

- спроектувати алгоритм масштабування, обробки та покращення зображень;

- здійснити програмну реалізацію методу;

- провести тестування програмного продукту.

- проаналізувати та дослідити отримані результати.

Для досягнення мети використано теоретичні та емпіричні та методи дослідження, а саме:

а) теоретичні методи:

– абстрагування – один з важливих методів, що дозволяє відкинути деякі несуттєві параметри (від абстрагування напряду залежить ефективність розроблюваної моделі);

– аналіз та синтез – декомпозиція моделі на прості складові, виявлення певних зв'язків поміж компонентами, а також синтез цих структурних елементів у єдине ціле;

– формалізація – представлення розроблюваної моделі у вигляді програмного коду;

б) емпіричні методи:

– спостереження (темою роботи є розробка методу та програмного засобу масштабування зображень, але для того, щоб виділити корисні ознаки, що повинні бути імплементовані у розроблюваних рішеннях, потрібно провести спостереження уже існуючих рішень, а також визначити їх властивості та зв'язки, що виникають між ними);

– експеримент (під час дослідження існуючих аналогів слід відтворити умови, що необхідні для аналізу імплементованих алгоритмів; пізніше цей же метод буде використано для аналізу ефективності результативного алгоритму, який розроблено та імплементовано у ході роботи).

Наукова новизна отриманих результатів:

– спроектовано алгоритм масштабування та покращення зображення без використання нейронних мереж;

– вперше розроблено метод масштабування зображення, який не використовуватиме величезну кількість ресурсів пристрою.

Практична цінність отриманих результатів полягає у розробці ефективного методу масштабування зображень, який не потребуватиме величезної кількості оперативної та відеопам'яті. Масштабовані зображення можуть бути використанні

для подальшої професійної обробки, публікації в мережі Інтернет або для друку на листі формату А4.

Достовірність та обґрунтованість отриманих результатів підтверджується використанням у процесі дослідження таких прийомів:

- перевірка нових рішень, ідей експериментальними дослідженнями за допомогою уже відомих процедур проектування та тестування;
- доведення працездатності та функціональної придатності розробленої програмної системи;
- наявність наукової публікації у рецензованому виданні.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Зображення високої роздільної здатності на сьогоднішній день є одним із найпопулярніших речей, які сьогодні потрібні людям. З кожним роком роздільна здатність екранів мобільної техніки стає все більшою, тому зростають вимоги до роздільної здатності та якості цифрового зображення, адже чим вища якість певного зображення, тим приємнішим і вагомим стає його перегляд для користувача. Оскільки якість цих зображень і відео швидко зростає, пропозиція та попит на ці продукти також стрімко зростають. Однак не завжди можливо досягти або створити найвищу якість зображень з технологічними обмеженнями, з якими стикаються під час належного процесу. Тому дуже часто використовуються алгоритми та методи масштабування та покращення цифрових зображень.

Масштабування зображення – це процес зміни роздільної здатності растрового зображення зі збереженням співвідношення його сторін, а також із збереженням або відновленням інформації, яка у ньому міститься. В цей процес також входить усування шумів, збільшення контрастності та нормалізація кольорової моделі зображення та його динамічного діапазону.

Дуже часто для покращення роздільної здатності використовують методи масштабування із використанням GAN (Generative Adversarial Networks). Завданням GAN є генерування зображення з «нуля», тобто використовуючи підібрані датасети, які дуже часто налічують в собі сотні мільйонів зображень, за допомогою яких цей GAN навчається.

GAN складається з двох компонентів: генератора та дискримінатора. Їхня особливість у тому, що вони функціонують, як конкуренти. Генератор здебільшого схожий на злодія, завданням якого є копіювання та створення реалістичних даних,

щоб ввести дискримінатора в оману. Генератор хоче обійти численні перевірки, які виконуватиме дискримінатор. У той же час, роль дискримінатора полягає в тому, щоб знаходити та виловлювати штучно згенеровані генератором дані. Дискримінатор намагається вловити аномалії та похибки генератора для того, щоб мати змогу виявити помилкові зображення, які були створені генератором.

Генератор приймає вхідний зашумлений сигнал і намагається створити з нього реалістичні зображення. Незважаючи на те, що спочатку він зазнає безліч невдач, він поступово навчиться робити якісніші та переконливіші зображення. Після цього дискримінатор намагається визначити, які зображення є справжніми, а які підроблені. Отже, дискримінатор та генератор конкурують між собою до тих пір, поки не буде досягнуто точки «ідеальності».

Згідно зі статтею [6], виявлено, що на відміну від генератора, для тренування дискримінатора використовують такі вихідні зображення:

- справжні дані, наприклад реальні зображення людей, тварин і т.д., які дискримінатор використовуватиме, як правдиві;
- дані, які згенерував генератор. Ці дані дискримінатор використовуватиме, як оманливі або фальшиві.

1.2 Аналіз існуючих методів та програмних засобів масштабування зображення та їх змістовний опис

1.2.1 Real-ESRGAN

Одним із основних програмних засобів масштабування зображення є Real-ESRGAN (Real Enhanced Super-Resolution Generative Adversarial Networks). Real-ESRGAN є покращеною та розширеною версією ESRGAN. Дана модель, була навчена

на наборі даних DIV2K (бікубічно зменшених зображеннях) на фрагментах зображення розміром 128 x 128, а також має відкритий вихідний код.

Генератор Real-ESRGAN являє собою серію блоків згорткової мережі з комбінацією з'єднань залишкових та щільних шарів (поєднання ResNet та DenseNet)

Основним завданням Real-ESRGAN є відновлення та покращення зображень із низькою роздільною здатністю шляхом їх масштабування та зменшення шуму. У роботі [7] описано тренування Real-ESRGAN та детальний опис його структури.

Найчастіше Real-ESRGAN використовують для покращення відео або анімаційних фільмів, а також відновлення втрачених частин фотографій. На версії ESRGAN існує графічний інтерфейс Upscale (рисунок 1.1). Основною вимогою стабільної роботи цього програмного засобу є наявність NVIDIA Cuda або Vulkan, для яких потрібна сучасна відеокарта та велика кількість відеопам'яті.

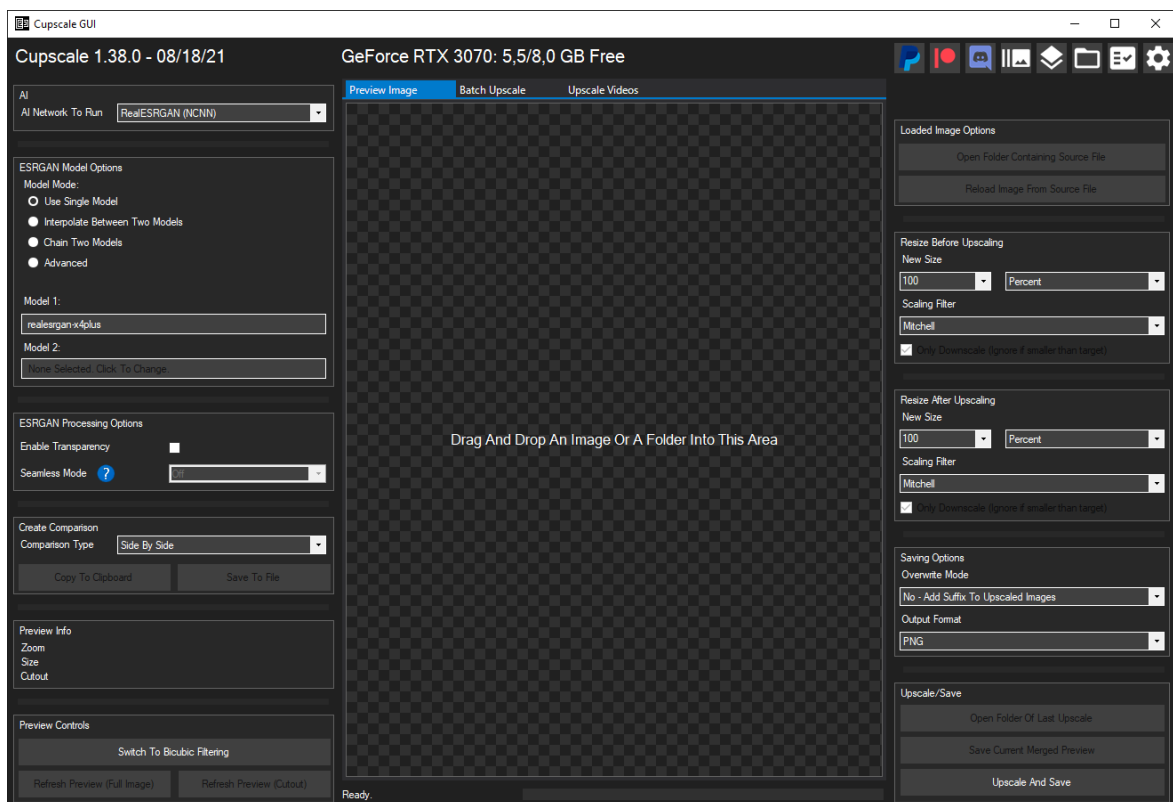


Рисунок 1.1 – Cupscale GUI

Результат масштабування зображення за допомогою моделі Real-ESRGAN та порівняння результату з вихідним зображенням показано на рисунку 1.2.



Рисунок 1.2 – Результат роботи Real-ESRGAN

1.2.2 Gigapixel AI

Також одним із найпопулярніших програмних методів для масштабування цифрових зображень є Topaz Gigapixel AI. Даний інструмент використовує штучний інтелект Gigapixel AI та дозволяє масштабувати зображення у декілька разів (до шестикратного збільшення).

Для отримання кращого результату для різних зображень використовують різні способи їх обробки. Для Gigapixel AI пропонує такі моделі штучного інтелекту:

- оригінальна модель штучного інтелекту (Standart);
- для архітектурних зображень або зображень з великою кількістю прямих ліній (Lines);

- для нефотографічних зображень, таких як анімація, комп'ютерна графіка та живопис (Art & CG);
- для невеликих файлів JPEG або зображень із серйозними артефактами стиснення (Very Compressed);
- призначена для отримання максимальної деталізації із зображень із низькою роздільною здатністю (Low Resolution);
- для покращення деталей маленьких обличч, наприклад, у масових сценах (Face Recovery).

Одними з основних недоліків даного методу масштабування зображення є надмірне використання ресурсів комп'ютера та обмежений функціонал у пробній версії програмного засобу.

Функціонал та результат роботи Gigapixel AI показано на рисунку 1.3.

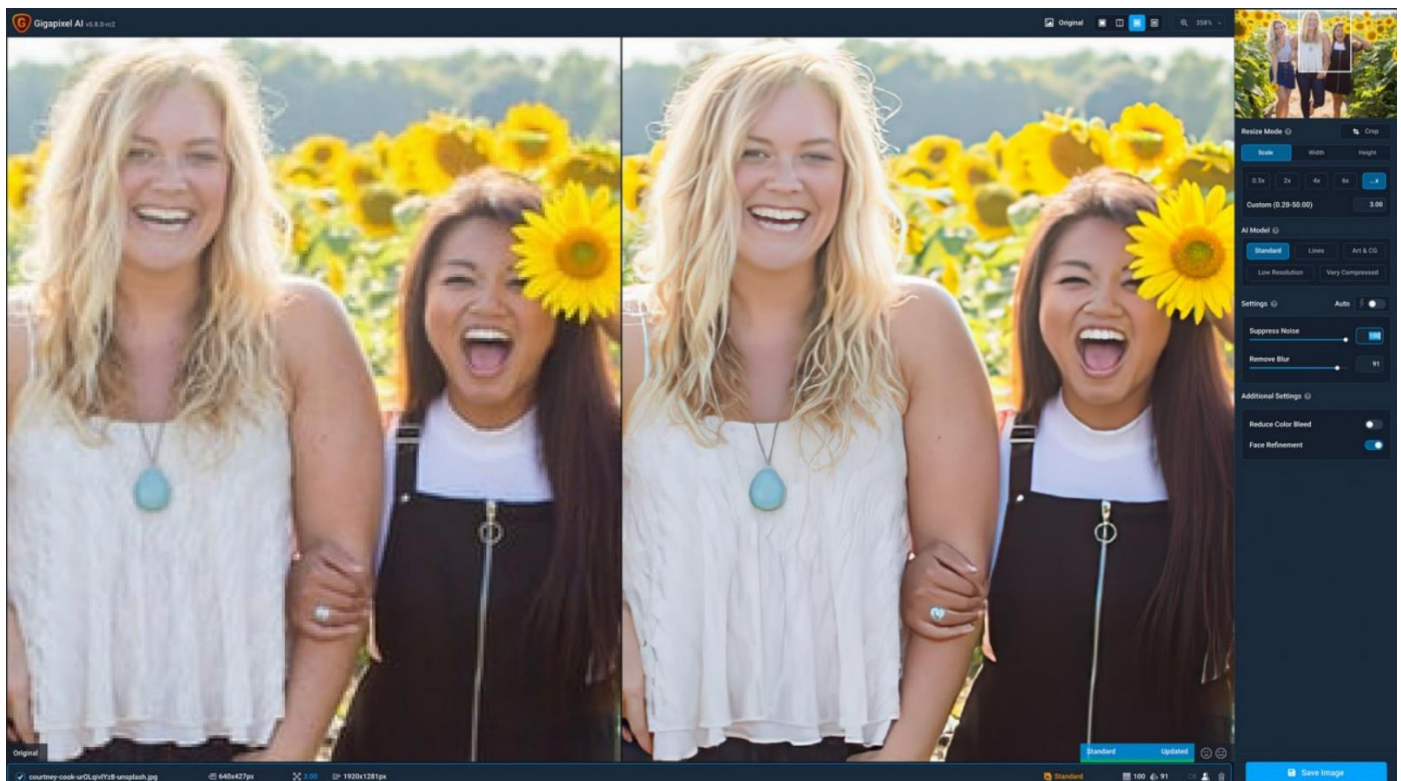


Рисунок 1.3 – Gigapixel AI

1.2.3 LapSRN

Під час обробки зображень дуже часто використовують піраміди, які є уявленням зображення з декількома різними роздільними здатностями. Такий метод дозволяє моделювати зображення з різною роздільною здатністю, від вихідного до дуже грубого зображення. Піраміда дозволяє алгоритму обробки зображення працювати від деталей до грубих. Цей інструмент використовується, зокрема, для сегментації зображень. Загальна структура піраміди показана на рисунку 1.4.

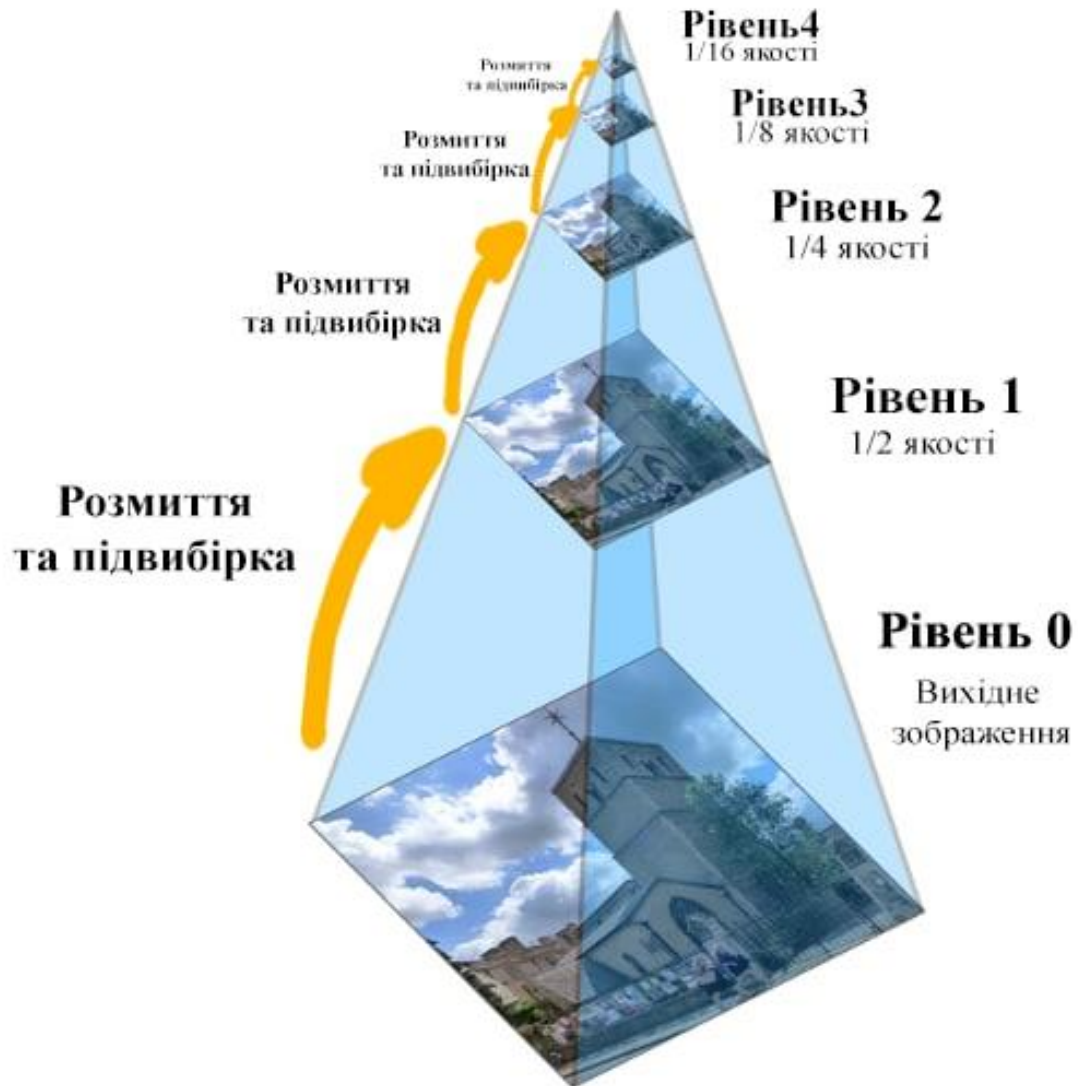


Рисунок 1.4 – Структура піраміди

Згідно з [9], одними з основних видів побудови піраміди зображення є:

- Гаусівський;
- Лапласа.

Під час використання піраміди Гауса отримані зображення модифікуються з використанням середнього значення за Гаусом (розмиття за Гаусом), після чого зменшуються. Кожен із пікселів містить локальне середнє значення, яке відповідає сусідньому пікселю нижнього рівня піраміди. Цей метод дуже часто використовується під час синтезу текстур [10].

Піраміда Лапласа є дуже схожою на піраміду Гауса, але їх основна відмінність полягає у тому, що вона розраховується шляхом збереження різниці з розмитим зображенням між кожним із рівнів. Лише найменший рівень не є різницею, тому його використовують для відновлення зображення з високою роздільною здатністю, використовуючи різницю між кожним рівнем. Цей метод часто використовують при стисканні зображень.

LapSRN (The Laplacian Pyramid Super-Resolution Network) використовує піраміду Лапласа для покращення якості зображення. Даний метод використовує дві гілки: гілку ознак (верхня гілка) та гілку відновлення зображення (нижня гілка) [10], які зображено на рисунку 1.5.

Піраміда Лапласа використовується десятиліттями. Її називають пірамідою Лапласа, тому що на гілці вилучення ознак виводиться одне залишкове зображення на кожному рівні.

На відміну від однокрокової підвищення дискретизації, мережа поступово реконструює залишки піддіапазонів зображень високої роздільної здатності на кількох рівнях піраміди.

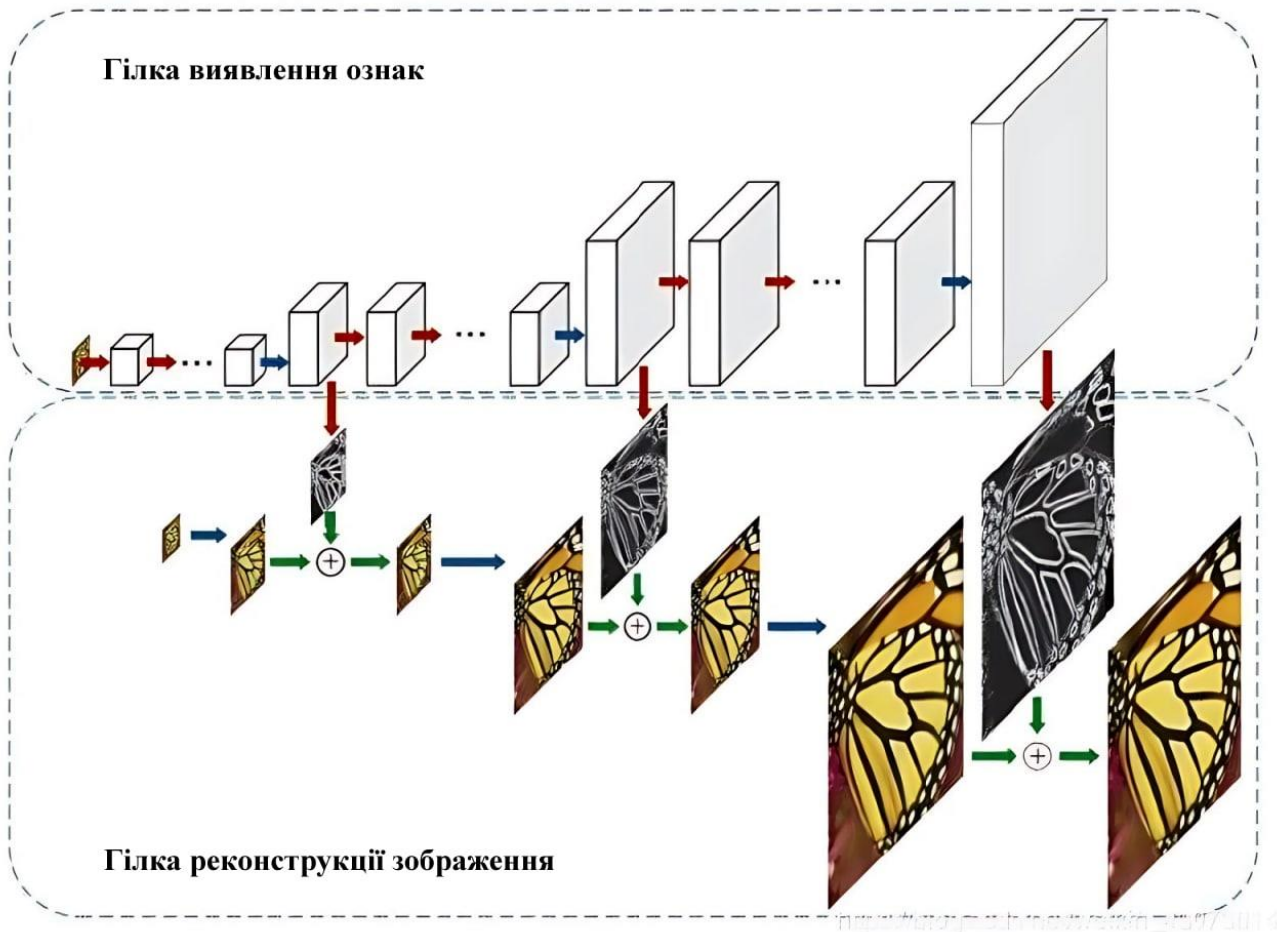


Рисунок 1.5 – Гілка виявлення ознак та гілка реконструкції зображення

1.2.4 Remini AI та GFPGAN

Нейронні мережі Remini AI та GFPGAN (Generative Facial Prior Generative Adversarial Networks) використовуються для усунення розмиття зображень, відновлення та покращення деталей, відновлення старих знімків. Основною відмінністю Remini AI та GFPGAN від інших вищезгаданих методів для масштабування зображень є те, що вони використовують навчені дані для покращення обличчя. Дані моделі використовують подібні алгоритми дій, тому для прикладу розглянуто алгоритм роботи GFPGAN.

GFPGAN використовує модуль для видалення деградації (U-Net), завданням якого є усунення деградації із вихідного зображення та витягування прихованих функцій. Згідно з [8], було виявлено, що цей модуль, зокрема, виділяє 2 типи функцій: приховані функції для відображення вхідного зображення з найближчим прихованим кодом StyleGAN2 та просторові функції з різною роздільною здатністю для модуляції функцій StyleGAN2.

Після цього навчена модель StyleGAN2 використовується у якості генеративної процедури обличчя. Між GAN і DRM приховані функції перетворюються кількома багаторівневими перцептронами (комп'ютерними моделями сприйняття інформації мозком) у вектори стилів. Далі ці вектори використовують для подальшої модуляції кінцевого результату.

GFPGAN дуже часто використовують разом з Real-ESRGAN, який допомагає покращити та відновити решту зображення, на якому обличчя відсутні (або іншими словами «non-face» ділянки).

На рисунку 1.6 зображено результати роботи Remini AI та GFPGAN, а також їх порівняння цих результатів зі збереженням оригінального співвідношення сторін.

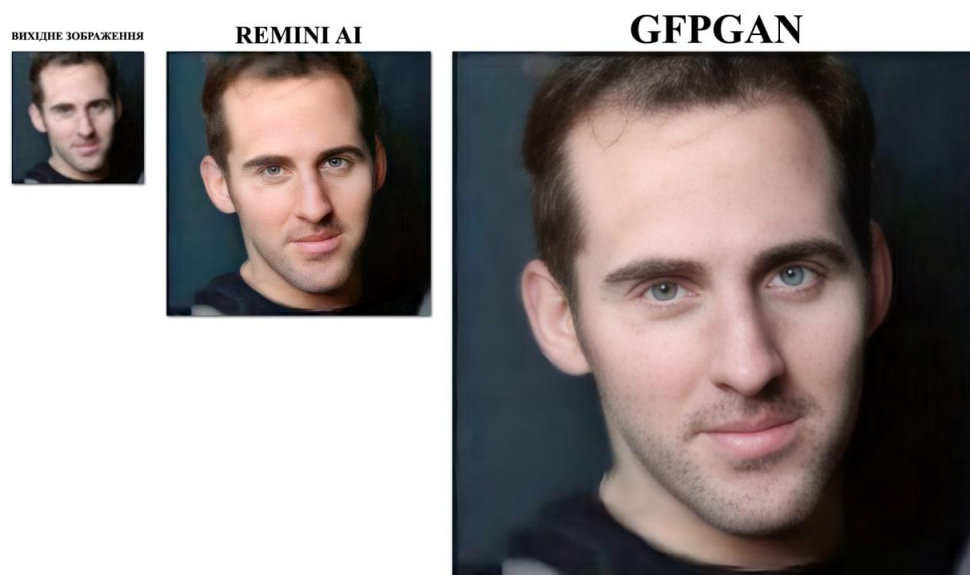


Рисунок 1.6 – Порівняння результатів роботи Remini AI та GFPGAN

1.2.5 EDSR

EDSR (Enhanced Deep Super-Resolution network) розроблювалась Бі Лімом та Санджюном Соном, як альтернатива для існуючих згорткових нейронних мереж. Дана використовує архітектуру блоку ResBlock для більш ефективного навчання моделі та розроблялась як розширена згорткова мережа для надзвичайно високої роздільної здатності одиночного зображення (Single Image) та вміщувала нову структуру ResNet блоків (рисунок 1.7).

Загальна архітектура мережі EDSR (c) у порівнянні з оригінальною (a) та SRResNet (b) показана на рисунку 1.8.

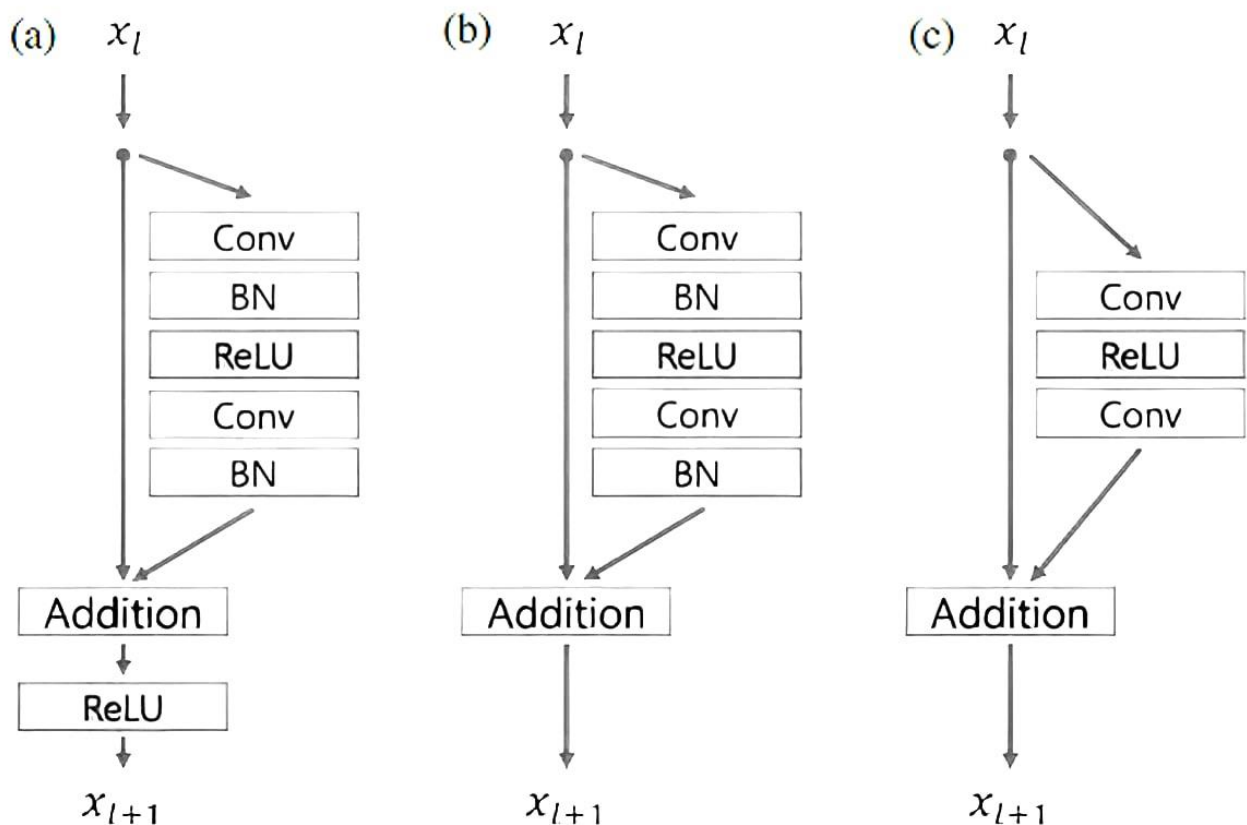


Рисунок 1.7 – Порівняння архітектур залишкових блоків

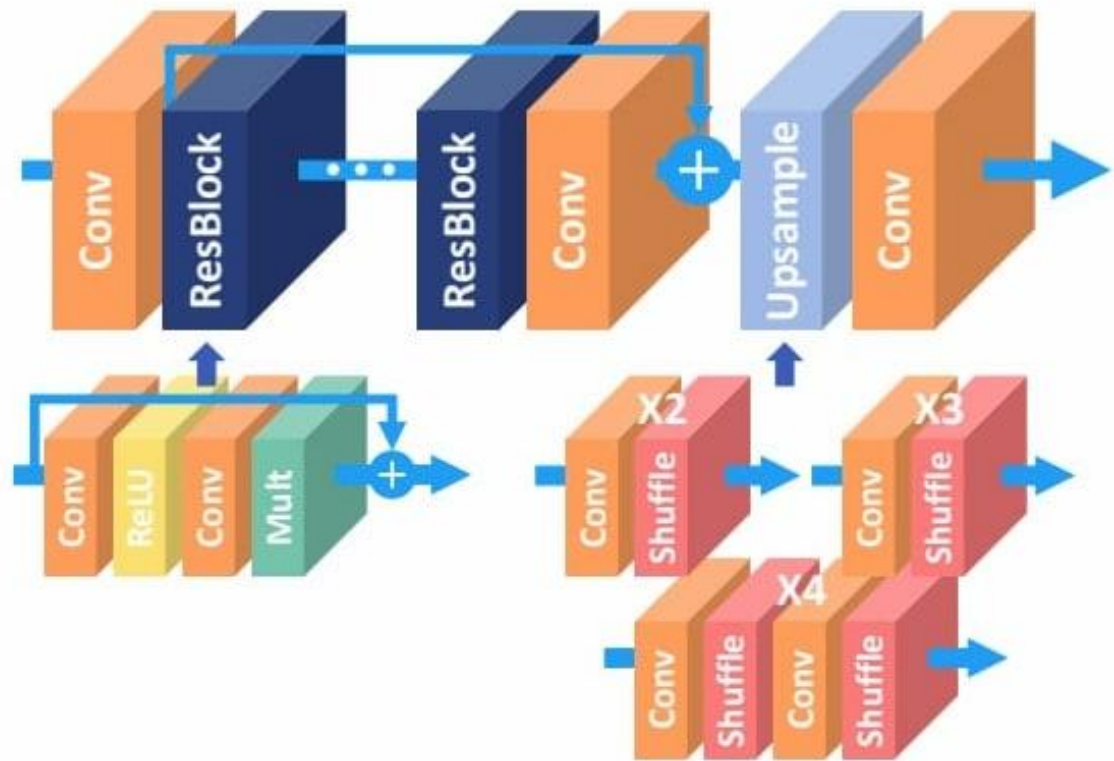


Рисунок 1.8 – Архітектура EDSR

1.2.5 Adobe Photoshop

Adobe Photoshop є одним із найвідоміших програмних продуктів, які мають можливість підвищувати роздільну здатність (рис. 1.9). Adobe Photoshop — це растровий графічний редактор, створений і розповсюджений Adobe Inc. для використання в операційних системах Windows і macOS [12].

Цей інструмент використовує один із SR-алгоритмів, і однією з його особливостей є здатність покращувати можливості розділення зображення. Крім того, плагіни можна використовувати для розширення набору доступних функцій.

Плагіни надають нові або вдосконалені функції, створюються та публікуються незалежно. Через його складність і широкий спектр функцій недоліки цього

програмного забезпечення можна пов'язати з типом ліцензії та високою ресурсозатратністю.

Існує безкоштовний пробний період для Photoshop, який є умовно-безкоштовним і для деяких функцій потребує реєстрації.

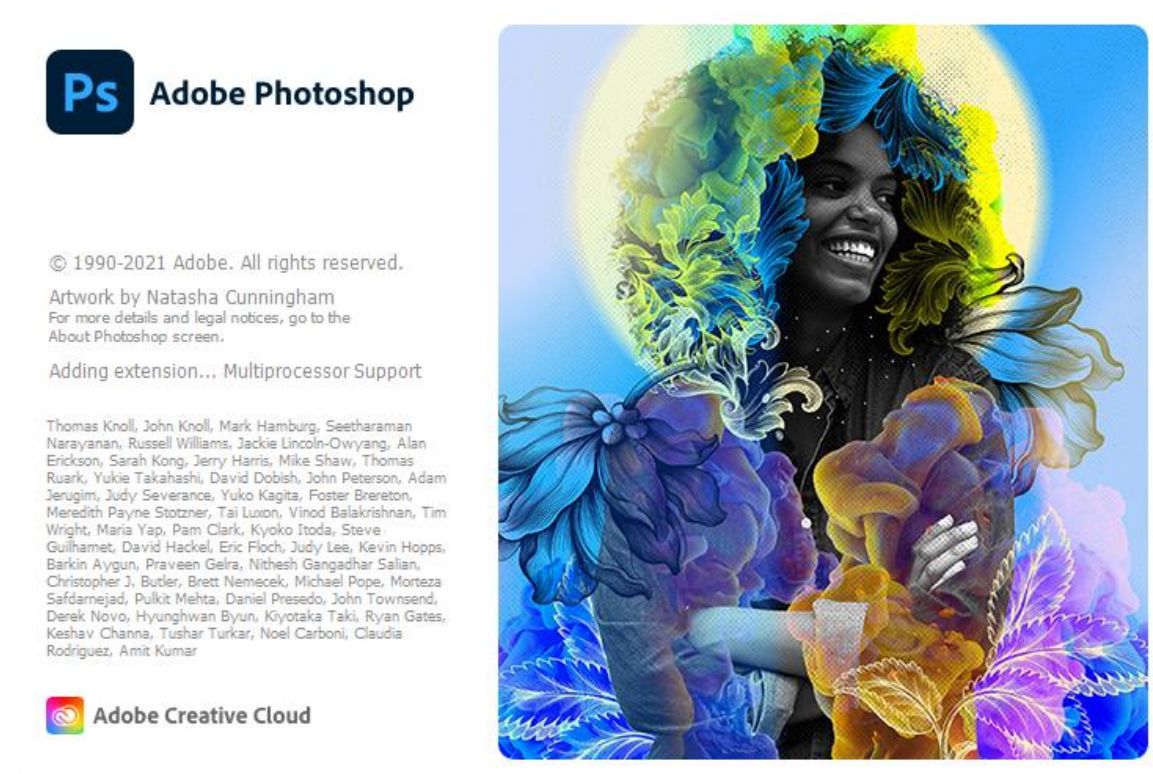


Рисунок 1.6 – Adobe Photoshop

1.3 Методологічні підходи до вирішення задачі масштабування зображення

Використання методів масштабування зображення, які використовують генеративні змагальні мережі (GAN) потребує велику кількість оперативної та відеопам'яті, що є основним і вагомим недоліком при використанні нейронних мереж для масштабування. Тому метод, що розроблюється буде використовувати інтерполяцію з подальшою обробкою зображення з метою відновлення деталей,

усунення шумів та розмиття, нормалізація кольорової моделі зображення, а також покращення його контрастності та динамічного діапазону.

Для розробки програмної системи буде використано монолітну архітектуру. Монолітна архітектура - це традиційна модель програмного забезпечення, яка є єдиним модулем, що працює автономно і незалежно від інших додатків. Моноліт часто називають щось велике і неповоротке, і ці два слова добре описують монолітну архітектуру для проектування ПЗ.

На відміну від мікросервісної архітектури монолітну архітектуру доречно використовувати при відсутності великої кількості сервісів і при наявності єдиної бази коду. Це допоможе у подальшій розробці, налагодженні та виявленні проблем при розробці програмної системи, при її програмній реалізації та проведенні тестування готового продукту.

При використанні готових продуктів проявляється залежність цих продуктів від системних вимог, у разі недостатності яких програмний засіб перестає функціонувати або видає збій при встановленні чи під час роботи. Також дуже часто виникає проблема із обробкою зображення, при якій на одну сесію обробки зображення при недостатньо сильному апаратному забезпеченні витрачається багато часу. Наприклад, на обробку зображення за допомогою нейронної мережі ESDR з використанням інтегрованої відеокарти та процесору Intel Core i3 сьомого покоління пішло близько восьми хвилин.

Даних проблем слід уникнути, тому прийнято рішення створити певну альтернативу цим методам і розробити власний метод, який буде уникати цих проблем. Тому замість використання AI буде реалізовано власний метод з використання інтерполяційних алгоритмів обробки зображення.

1.4 Висновки. Постановка задачі

Одними з основних проблем при використанні існуючих методів є:

- використання надмірної кількості ресурсів;
- високі вимоги до апаратних компонентів;
- втрата деталей у деяких випадках (таких як текст і т.д.).

Таким чином з'являється потреба у створенні методу масштабування зображення, який задовольнятиме наступні критерії:

- швидкодія;
- низька ресурсовитратність;
- помірні вимоги до апаратних пристроїв;
- простота у використанні.

Актуальність теми полягає у тому, що дуже часто користувачі або розробники стикаються із растровими зображеннями низької або недостатньо високої якості, тому дуже часто вони користуються програмними продуктами, які дозволяють покращити якість та масштабувати зображення.

– Тема масштабування зображення актуальна на протязі декількох десятків років і до сих пір розробники знаходять нові рішення та методи щодо збільшення масштабу зображення із меншою кількістю втрат якості

– Більшість наявних методів використовують нейронні мережі, що призводить до зайвих навантажень на пристрій, тому виникає потреба у розробці нового алгоритму, ґрунтуючись на проблемах, які є присутніми у вже існуючих методах.

Об'єкт дослідження – масштабування зображень.

Предмет дослідження – методи, алгоритми та програмні засоби масштабування растрових зображень.

Мета дослідження – розробка методу масштабування зображення, який не буде використовувати велику кількість ресурсів та буде поєднувати збільшення роздільної здатності зображення із покращенням його кольорового тону.

Відносно мети можна сформулювати наступні завдання дослідження:

- проаналізувати предметну область масштабування зображень;
- проаналізувати наявні методи та програмні засоби масштабування зображень та дослідити їх недоліки;
- створити алгоритм роботи методу;
- програмна реалізація власного методу, завданням якого буде компенсування недоліків існуючих методів;
- провести тестування та порівняти результати із наявними методами.

Таким чином, у даному розділі було проаналізовано предмету область, розглянуто та проаналізовано наявні методи та програмні засоби масштабування зображень, описані їх недоліки, запропоновано власний метод масштабування, а також виконана постановка задачі для подальшого дослідження.

2 КОНЦЕПЦІЇ, МОДЕЛІ ТА МЕТОДИ ВИРІШЕННЯ ЗАДАЧІ

2.1 Концепції масштабування зображень

Більшість традиційних методів масштабування зображень використовують нейронні мережі. Завданням нейронних мереж є відновити зображення із вже готового датасету, на якому нейронна мережа була навчена. Основною проблемою є те, що даний процес займає велику кількість часу та відеопам'яті, а також зазнає втрат у дрібних деталях. Тому дуже часто доцільніше використовувати інтерполяцію.

Інтерполяція – це процес визначення значень функції при позиції, розташованих між його зразками. Він досягає цього процесу шляхом встановлення безперервної функції через дискретні вхідні зразки. Це дозволяє оцінювати вхідні значення у випадкових позиціях у вхідних даних, а не лише в тих, що визначені в точках вибірки. У той час як вибірка генерує сигнал нескінченної смуги пропускання з сигналу з обмеженою смугою, інтерполяція відіграє протилежну роль: вона зменшує смугу пропускання сигналу, застосовуючи до дискретного сигналу фільтр низьких частот. Тобто інтерполяція реконструює сигнал, втрачений у процесі дискретизації, згладжуючи вибірки даних за допомогою функції інтерполяції.

Процес інтерполяції є однією з основних операцій обробки зображень. Якість зображення багато в чому залежить від використовуваної техніки інтерполяції. Методи інтерполяції діляться на дві категорії: методи детермінованої та статистичної інтерполяції. Різниця полягає в тому, що методи детермінованої інтерполяції припускають певну мінливість між точками вибірки, наприклад лінійність у випадку лінійної інтерполяції. Методи статистичної інтерполяції апроксимують сигнал шляхом мінімізації похибки оцінки. Цей процес наближення може призвести до того, що вихідні значення вибірки не відтворюватимуться.

Також буде доречно описати алгоритми роботи методів інтерполяції, їх особливості та їх структуру.

У таблиці 2.1 описано основні методи інтерполяції та їх властивості.

Таблиця 2.1 – Методи інтерполяції та їх властивості

Алгоритм інтерполяції	Властивість
Інтерполяція методом найближчого сусіда або округла інтерполяція	Є одним із найпростіших алгоритмів інтерполяції. Приймає округлене значення очікуваної позиції та знаходить найближче значення даних у позиції цілого числа.
Білінійна інтерполяція	Білінійна інтерполяція є узагальненням лінійної інтерполяції для функції із двома змінними. На відміну від алгоритму інтерполяції найближчого сусіда, білінійна інтерполяція виконує лінійну інтерполяцію по одній осі, а потім по іншій осі.
Інтерполяція співвідношенням площ	Використовує повторну дискретизацію з використанням співвідношень площ пікселів. Цей алгоритм використовують при зменшенні зображення, адже при збільшенні даний метод не відрізняється від інтерполяції найближчого сусіда.
Бікубічна інтерполяція	Бікубічна інтерполяція відрізняється від білінійної інтерполяції завдяки використанню 16 пікселів (4×4 пікселі), тоді як білінійна використовує 4 пікселі (2×2), в результаті чого можна отримати більш згладжений результат.

Кінець таблиці 2.1

Інтерполяція Ланцоша	Інтерполяція Ланцоша використовує 64 сусідні пікселі (8×8) і усереднює їх використовуючи функції sinc. Інтерполяція Ланцоша часто використовується при зменшенні дискретизації зображень, тобто при зменшенні їх роздільної здатності.
----------------------	--

Найпростішим алгоритмом інтерполяції з обчислювальної точки зору є метод найближчого сусіда, де для кожного інтерпольованого вихідного пікселя присвоюється значення найближчої точки вибірки у вхідному зображенні. Також дана техніка відома як алгоритм зміщення точки та реплікація пікселів. Ядро інтерполяції для алгоритму найближчого сусіда визначається наступною формулою (2.1).

$$h(x) = \begin{cases} 1 & 0 \leq |x| < 0.5 \\ 0 & 0.5 \leq |x| \end{cases} \quad (2.1)$$

Ядро алгоритму інтерполяції методом найближчого сусіда показано на рисунку 2.1.

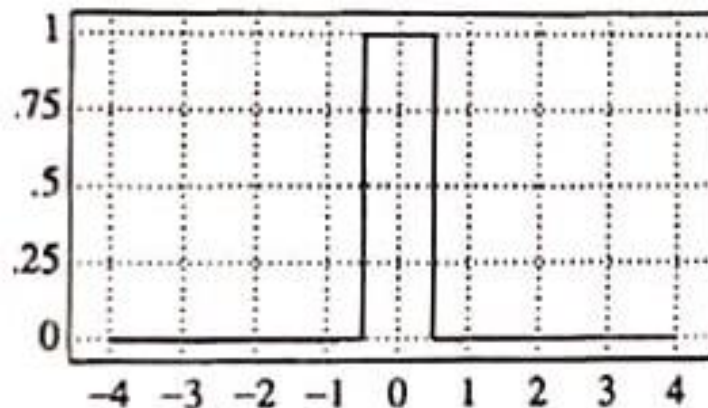


Рисунок 2.1 – Ядро інтерполяції методом найближчого сусіда

Лінійна інтерполяція — це метод першого ступеня, який пропускає пряму лінію через кожні дві послідовні точки вхідного сигналу.

У просторовій області лінійна інтерполяція еквівалентна згортанню вибірки вхідних даних із ядром [12], показаним у формулі (2.2).

$$h(x) = \begin{cases} 1 - |x| & 0 \leq |x| < 1 \\ 0 & 1 \leq |x| \end{cases} \quad (2.2)$$

Ядро алгоритму лінійної інтерполяції показано на рисунку 2.2.

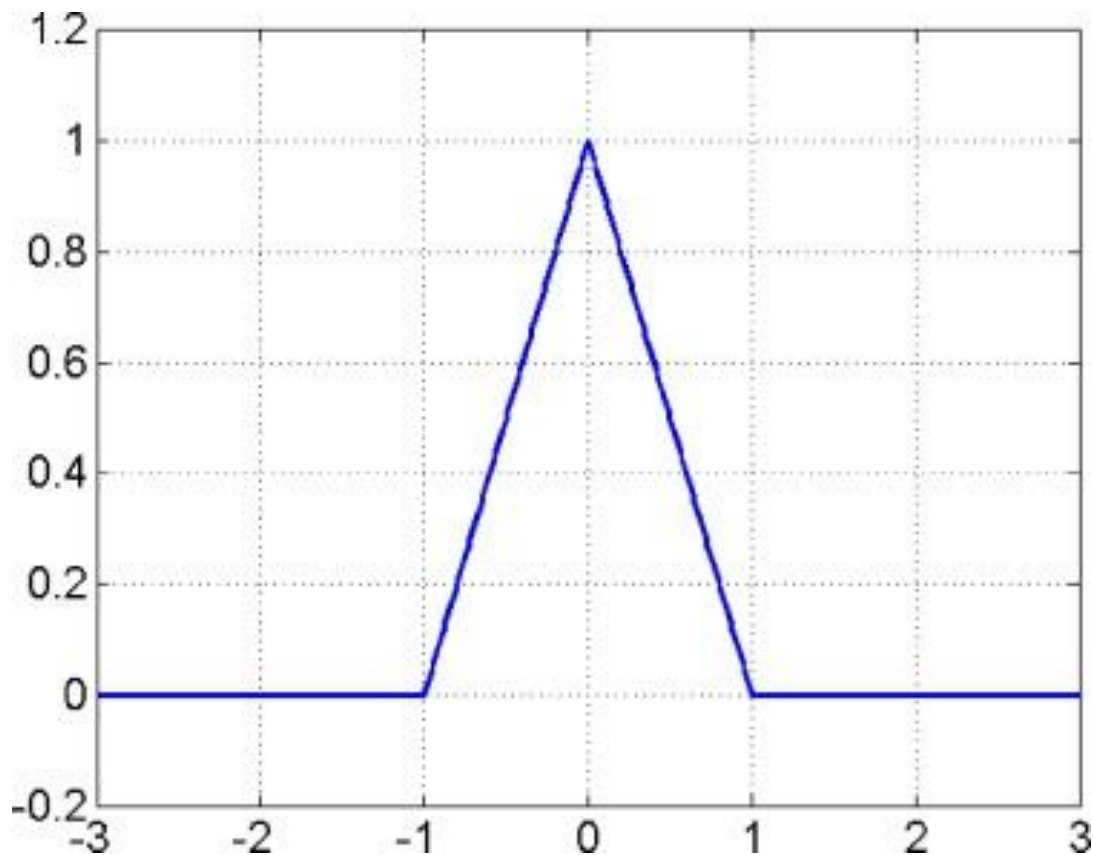


Рисунок 2.2 – Ядро лінійної інтерполяції

Кубічна інтерполяція – інтерполяція функції однієї змінної за допомогою поліному третього ступеня. Ядро методу кубічної інтерполяції складається з кусково-кубічних поліномів, які є визначеними на підінтервалах $(-2,-1)$, $(-1,0)$, $(0,1)$ і $(1,2)$. За

межами цього інтервалу ядро інтерполяції дорівнює нулю. Щоб отримати ядро кубічної інтерполяції, потрібно вирішити 8 лінійних рівнянь із 7 невідомими параметрами (2.3).

$$h(x) = \begin{cases} (a + 2)|x|^3 - (a + 3)|x|^2 + 1 & 0 \leq |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| < 2, \\ 0 & 2 \leq |x| \end{cases} \quad (2.3)$$

де a є один «вільним» параметром, який задає користувач. Для більшості зображень використовують різне значення параметру a , адже у різних випадках від вибору значення цього параметру залежить якість результату [13].

На рисунку 2.3 зображено ядро кубічної інтерполяції.

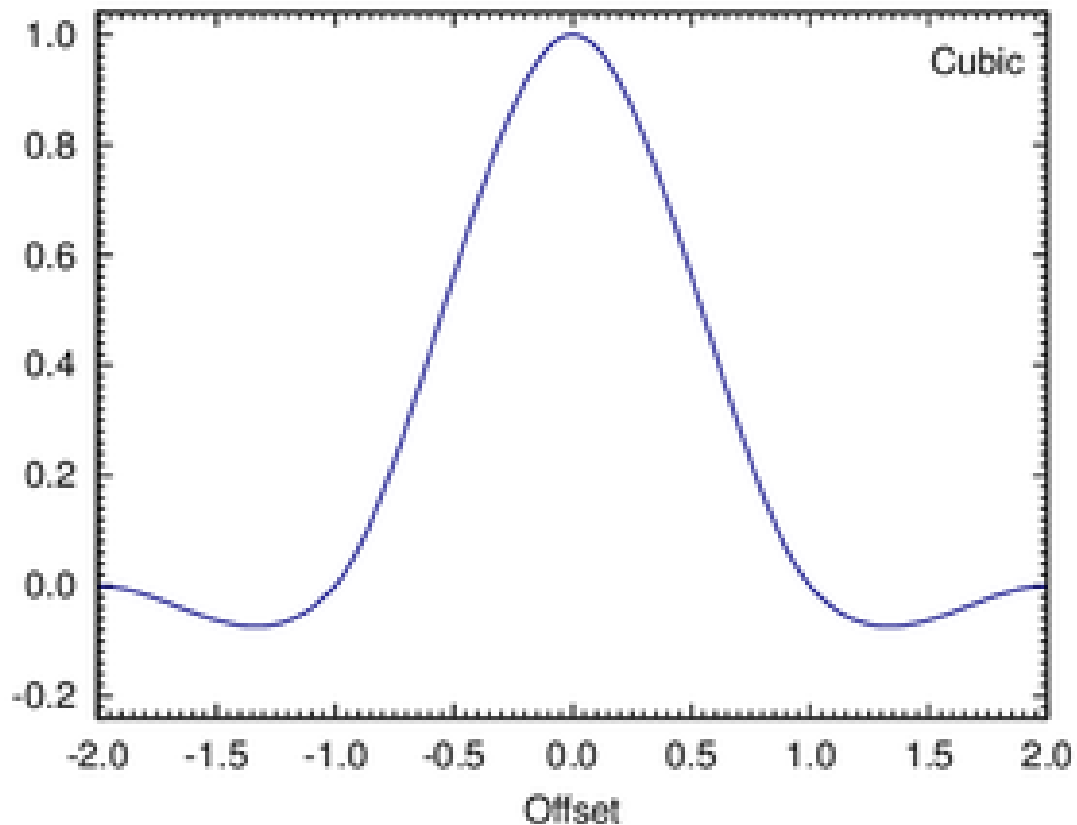


Рисунок 2.3 – Ядро кубічної інтерполяції

Інтерполяція Ланцоша використовує метод математичної обробки рядів даних. Застосовується або для інтерполяції функції між заданими точками, або як фільтр нижніх частот. Найчастіше застосовується у обробці оцифрованих зображень, зміни їх роздільної здатності або передискретизації [13].

Вплив кожної вхідної вибірки на інтерпольовані значення визначається ядром реконструкції фільтра $L(x)$, яке називається ядром Ланцоша.

$$L(x) = \begin{cases} \text{sinc}(\pi x) \text{sinc}(\pi x/a) & -a < x < a \\ 0 & \text{інакше} \end{cases}, \quad (2.4)$$

де sinc є нормалізованою функцією кардинального синусу.

Доречним буде замінити функцію кардинального синусу на функцію звичайного синусу, в результаті чого отримано рівносильну формулу (2.5)

$$L(x) = \begin{cases} 1 & x = 0 \\ \frac{a \sin(\pi x) \sin(\pi x/a)}{\pi^2 x^2} & -a \leq x < a, x \neq 0 \\ 0 & \text{інакше} \end{cases}, \quad (2.5)$$

де параметр a є додатним числом, наприклад $a = 2$, або $a = 3$

На рисунку 2.4 показано, яке поводить себе графік ядра інтерполяції Ланцоша при різних значеннях параметру a .

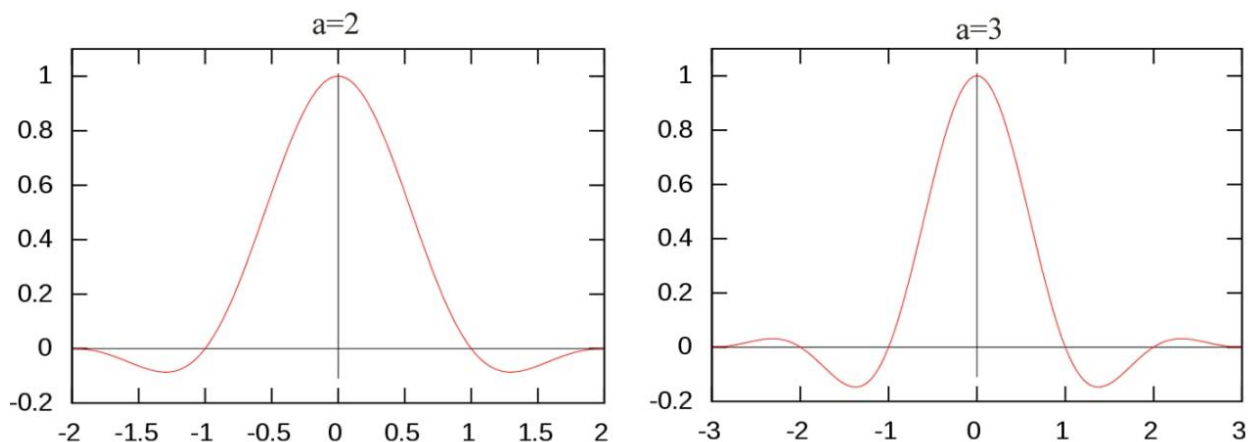


Рисунок 2.4 – Ядро інтерполяції Ланцоша при різних параметрах

Також на рисунку 2.5 показано графічне відображення роботи алгоритмів інтерполяції зображення.

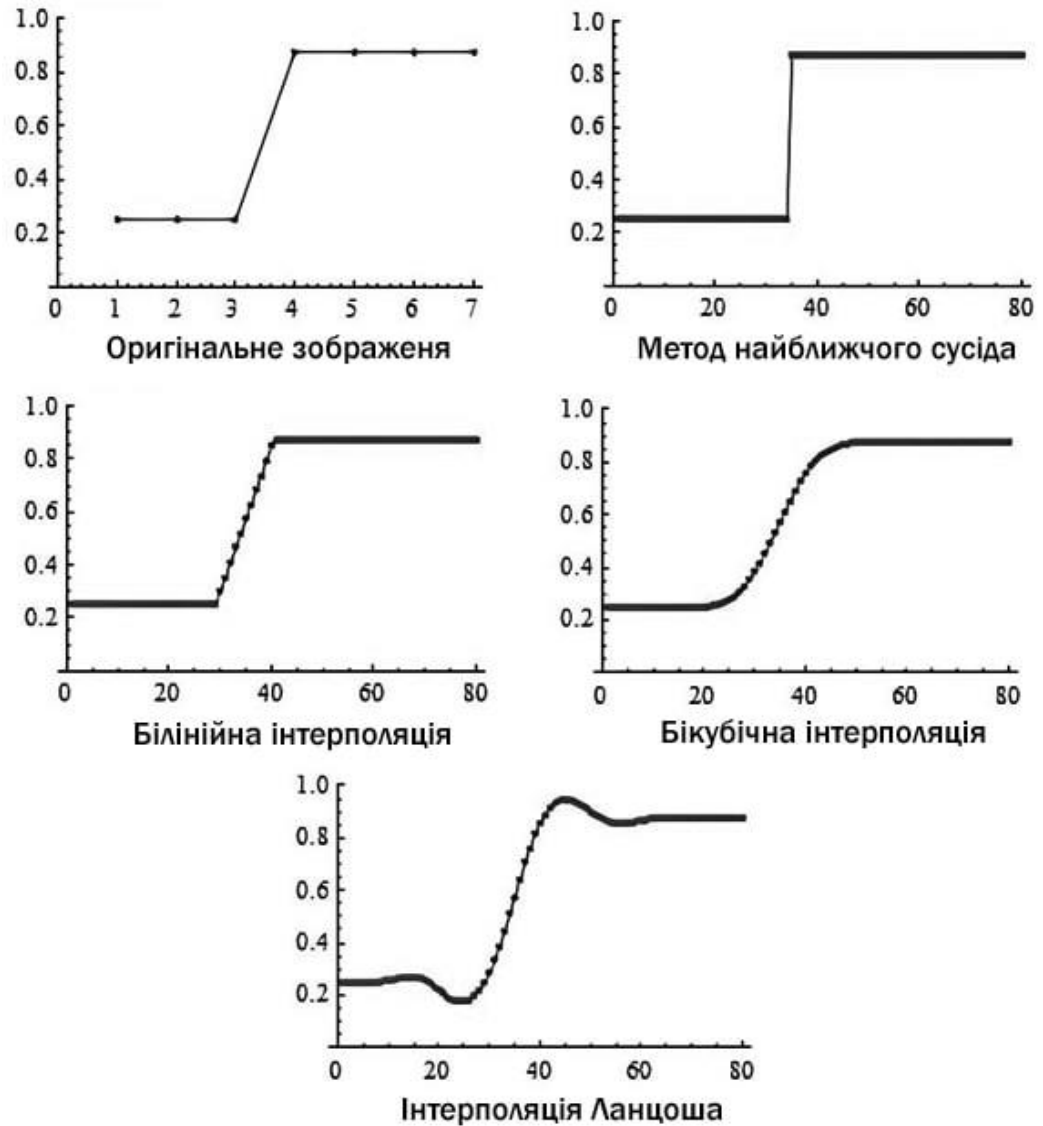


Рисунок 2.5 – Графічне відображення алгоритмів інтерполяції

Для реалізації власного методу масштабування зображення було вирішено використовувати білінійну інтерполяцію, а саме її варіацію, яка дозволить здійснити інтерполяцію зображення з точністю до біту (LINEAR_EXACT).

2.2 Моделі та методи масштабування зображення

На базі описаних у розділі 2.1 методів масштабування та покращення зображення розробимо програмну систему, яка дозволить отримати задовільний результат та уникати проблем.

Стандартні методи інтерполяції без подальшої обробки зображення видають результат середньої якості, тому при розробці буде застосовано

Для подальшої розробки методу та програмного засобу для масштабування цифрового зображення розроблено алгоритм, який допоможе при програмній реалізації методу. Даний алгоритм показано на рисунку 2.6.

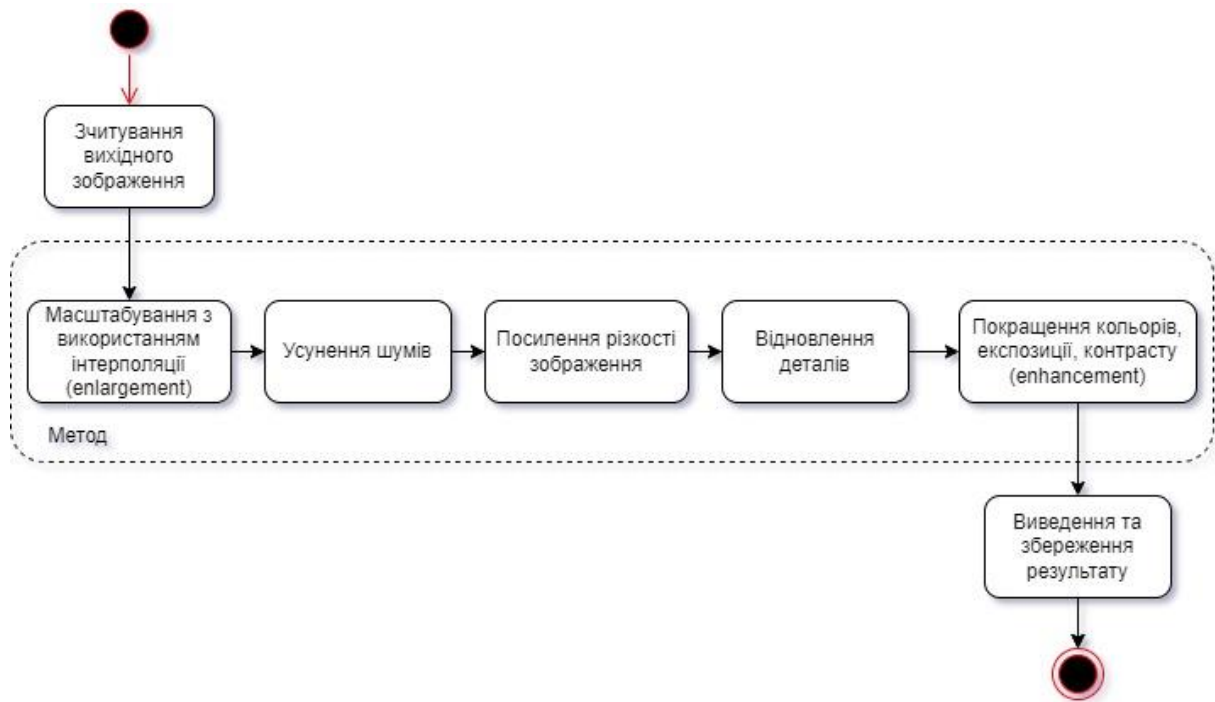


Рисунок 2.6 – Алгоритм роботи методу

У результаті інтерполяції зображення низької роздільної здатності можуть виникати артефакти зображення, такі як шум та втрата деяких деталей. Крім того деякі частини зображення можуть бути розмитими.

Усунення шумів є дуже важливою частиною зображення. Для цього процесу доречно використовувати білатеральний фільтр. Білатеральний фільтр замінює інтенсивність кожного пікселя середньозваженим значенням інтенсивності сусідніх пікселів. Це здійснюється завдяки поєднанню двох фільтрів Гауса: один із фільтрів працює у просторовій області, в той час як інший працює в області інтенсивності (або яскравості). Особливість білатерального фільтру у тому, що він дає змогу усунути шуми зі збереженням країв зображення, на відміну від розмиття медіаною.

Результат роботи білатерального фільтру можна побачити на рисунку 2.7.



Рисунок 2.7 – Результат роботи білатерального фільтру

Також важливим компонентом при обробці растрових зображень є використання ядра зображення. Ядро зображення – це невелика матриця, яку

використовують для застосування ефектів, подібних до тих, які можна знайти у Photoshop, наприклад розмивання, збільшення різкості, виділення країв. Дуже часто ядро зображення використовують у машинному навчанні для «виділення ознак» зображення, а саме для визначення найважливіших його частин. Використання ядра зображення також називають «згорткою».

У обробці зображень ядро – це двовимірна матриця чисел. Така матриця може мати різні розміри, але здебільшого використовують розмірні ядра 3x3. Матрицю ядра розміром 3x3 використовують для околиці 9 пікселів [14].

У таблиці 2.2 показано основні матриці ядра зображення, які використовуються при його обробці.

Таблиця 2.2 – Ядра зображення

Операція	Визначення	Ядро
Ідентичність	Показує ідентичне (тотожне) зображення. Таке ядро також називають «do-nothing» ядром.	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
Виділення країв	Виявляє краї всередині зображення. Дана техніка працює шляхом виявлення різких перепадів яскравості зображення та її розривів.	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
		$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$
Розмиття коробкою (середнє розмиття)	Замінює середнє значення кожного пікселя зображення середнім значенням усіх оточуючих його пікселів.	$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Кінець таблиці 2.2

Гаусове розмиття	Здійснює розмиття зображення за допомогою функції Гауса, яка створює розподіл значень навколо центральної точки. В результаті чого утворюється ядро, в якому пікселі поблизу центру роблять більший внесок у нове значення пікселів, ніж пікселі, розташовані далі.	$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$ $= \begin{pmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{pmatrix}$
Посилення різкості	Дозволяє посилити різкість зображення, що дозволить посилити різкість його країв.	$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$

При розробці методу масштабування зображення буде використано ядро посилення різкості, що допоможе відновити деталі зображення після використання білатерального фільтру.

Для покращення та вирівнювання контрастності буде доречним використання адаптивного вирівнювання гистограми (АНЕ). Дана модель відрізняється від звичайного вирівнювання гистограми тим, що він обчислює декілька гистограм окремих частин зображення. Після цього він використовує їх для перерозподілу значень освітленості зображення. Ця модель підходить для покращення локального контрасту та покращення чіткості країв у кожній області цифрового зображення.

Адаптивне вирівнювання гістограми має певну тенденцію посилення шуму у однорідних областях цифрового зображення. В тому випадку буде використано адаптивне вирівнювання гістограми з обмеженим контрастом (CLAHE) [15].

При використанні CLAHE посилення контрасту в околицях заданого піксельного значення визначається завдяки нахилу функції перетворення. Це пропорційно нахилу кумулятивної функції розподілу сусідства (CDF), а також значенню гістограми при цьому значенні пікселя. CLAHE дозволяє обмежити посилення завдяки обрізанню гістограми на попередньо визначеному значенні до обчислення CDF. Це дозволить обмежити нахил CDF та функцію перетворення. Обмеження відсікання, тобто значення, при якому обрізається гістограма, залежить від нормалізації гістограми і, таким чином, від розміру сусідньої області.

Перерозподіл призведе до того, що деякі області знову перевищать обмеження кліпу (область, затінена зеленим кольором на рисунку 2.8), після чого ефективно обмеження кліпу буде більшим за встановлене обмеження і точне значення якого залежить від зображення. Якщо це небажано, процедуру перерозподілу можна повторювати рекурсивно, поки надлишок не стане незначним.

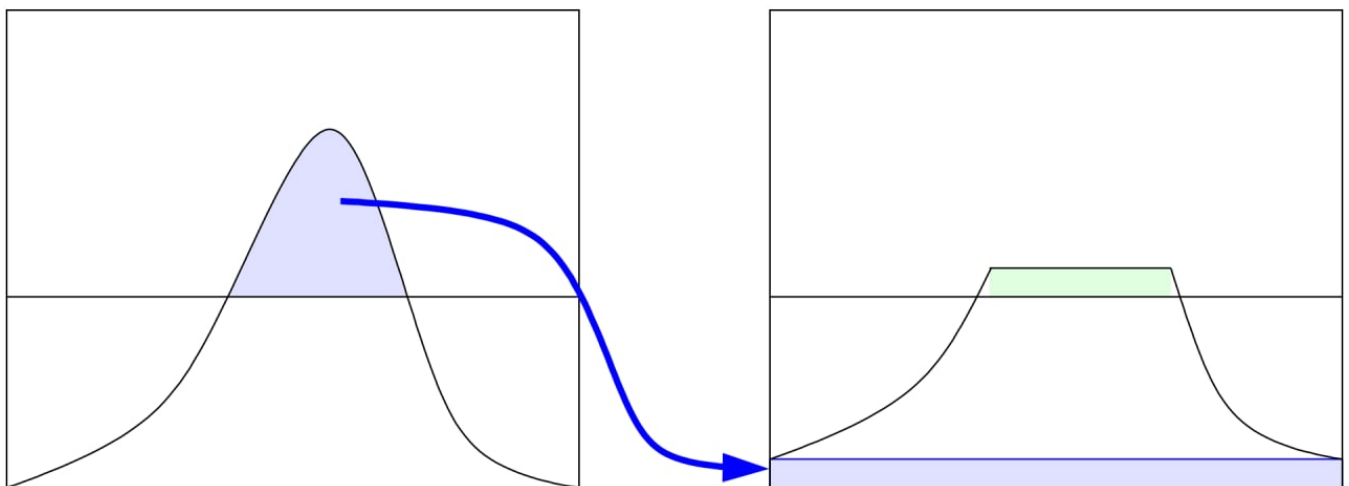


Рисунок 2.8 – Модель CLAHE

Для більшої продуктивності моделі CLANE буде використано перетворення колірному простору зображення у CIELAB або LAB. Дана колірна модель розглядає зображення використовуючи такі параметри: L^* – світлосила, а – відношення зеленого кольору до червоного та b^* – відношення синього кольору до жовтого (рисунок 2.9). Дана колірна модель дає змогу задати кольори без урахування певної особливості їх відтворення, тобто, коротше кажучи, є апаратно-незалежною.

На відміну від колірному простору RGB, при використанні якого починають із червоного кольору та додають різні відтінки червоного, зеленого та синього, LAB дозволяє керувати трьома різними каналами, що допомагає при задачах, які при використанні RGB займають більше часу або і зовсім є неможливими, наприклад при корекції кольору або при тонуванні.

На рисунку 2.10 показано порівняння колірних схем RGB та CIELAB.

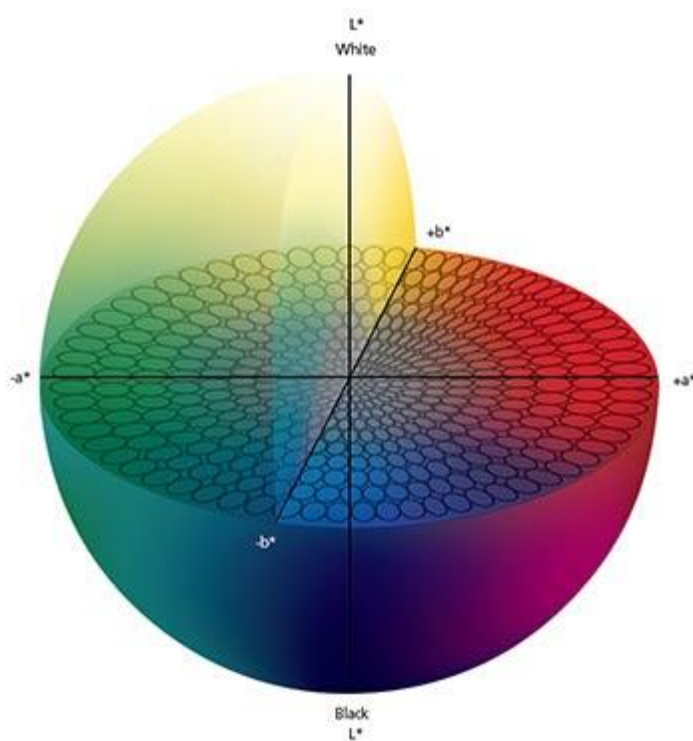


Рисунок 2.9 – Колірна модель CIELAB (LAB)

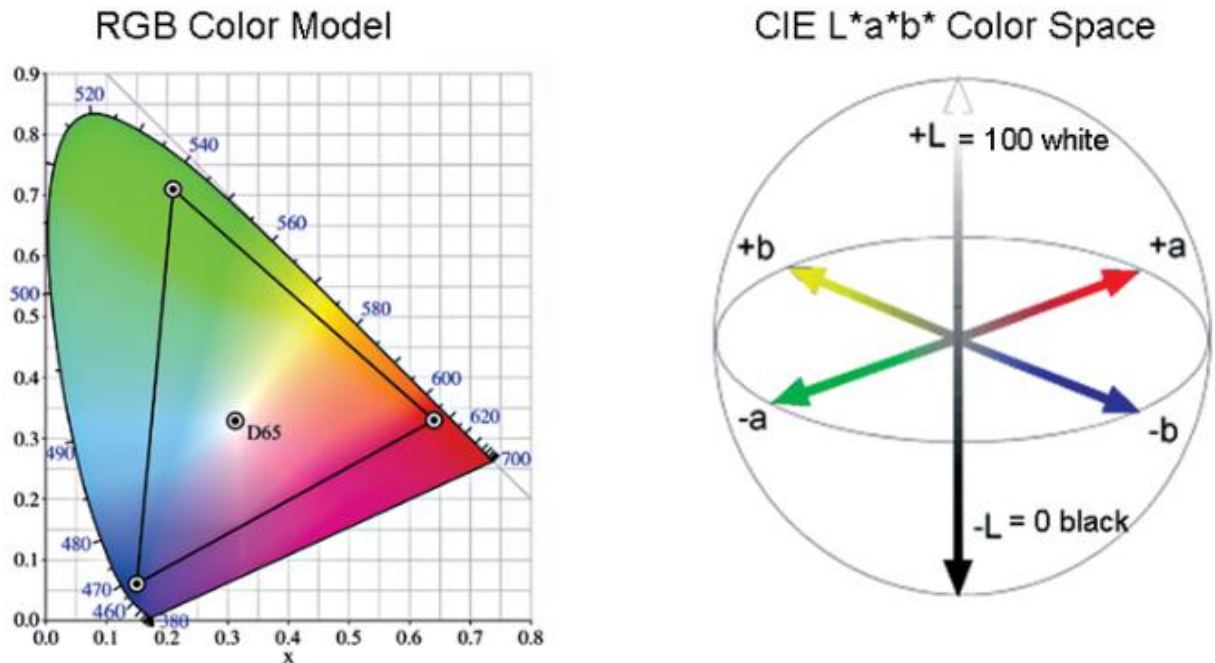


Рисунок 2.10 – Порівняння колірних схем RGB та LAB

L-канал буде використаний для застосування до нього CLAHE, що дозволить уникнути втрат у якості під час адаптивного вирівнювання гістограми.

2.3 Висновки

У розділі було здійснено аналіз моделей та методів масштабування зображення. Наприклад, один із основних компонентів масштабування є інтерполяція зображення, тому було розглянуто основні проблеми та переваги основних алгоритмів інтерполяції.

Також було розглянуто ядра інтерполяції, що дозволило більш детально оглянути структуру та функціонал кожного із алгоритмів, завдяки чому було вибрано один із них, який використовуватиметься при розробці власного методу.

Було розглянуто засоби для усунення шумів та артефактів растрових зображень, а саме білатеральний фільтр, а також ядра зображення та матриці, які вони використовують.

На основні розглянутих моделей розроблено алгоритм, який допоможе у подальшій розробці методу та програмного засобу з уникненням наявних проблем у існуючих методах.

Наступним етапом розробки є аналіз детальних вимог до програмного засобу, його проєктування та декомпозиція, підбір та обґрунтування засобів реалізації.

3 ТЕХНОЛОГІЯ РЕАЛІЗАЦІЇ МЕТОДУ МАСШТАБУВАННЯ ТА ПОКРАЩЕННЯ ЗОБРАЖЕНЬ

3.1 Аналіз вимог до програмного засобу

Після аналізу предметної області, розробленого методу та алгоритмів, потрібно сформулювати основні вимоги до програмного засобу, метою розробки якого є створення власного методу масштабування зображень. Розроблюваний метод повинен вирішувати наявні проблеми предметної області масштабування, тому можна виділити наступні характеристики програмного продукту:

- обробка зображення із мінімальними витратами часу та ресурсів;
- дотримання максимальної схожості із початковими зображенням;
- уникнення критичних помилок та таймаутів.

1) Класи користувачів та їх характеристики

Програмна система міститиме лише один клас користувачів. Користувач матиме можливість виконувати лише вхідні запити (решта роботи буде виконуватись на основі рішень алгоритму автоматично).

2) Середовище функціонування

Середовищем функціонування є програмний засіб, розроблений на монолітній архітектурі, тобто є монолітним застосунком, який не буде залежати від інших продуктів та функціонуватиме незалежно від них.

3) Характеристика системи

Обробник та система тимчасових файлів

Для контролю моментів запуску та завершення роботи з програмним засобом буде використано обробник подій.

Послідовності дія/відгук:

а) запуск обробника після запуску програмного продукту допоможе контролювати та відстежувати момент початку та завершення роботи користувача із програмним засобом, що допоможе у подальшій його роботі;

б) створення директорії для тимчасових файлів (ДТФ) допоможе у масштабуванні та обробці зображень, а саме зберігати їх при обробці та видаляти, якщо користувач вирішить не зберігати результати;

в) призначення ДТФ атрибуту «Прихований», що дасть можливість уникнути доступу користувача до цих файлів;

г) якщо директорія для збереження результатів відсутня, то програмний засіб створює її;

д) тимчасове збереження результатів роботи методу, що допоможе при обробці широкого динамічного діапазону.

Збільшувач та поліпшувач

Завантажене користувачем зображення буде оброблятися за допомогою масштабування або збільшення (enlarge) та поліпшення (enhance).

Послідовність дія/відгук:

а) метод масштабування зображення прийматиме зображення, сторони якого не більші за 2048x2048 пікселів, що допоможе уникнути проблем при обробці растрових зображень;

б) у якості вхідного зображення приймаються файли з розширенням .jpg та .png, які є найбільш розповсюдженими;

в) якщо процес масштабування та поліпшення зображення завершився, то результат зберігається у форматі PNG, щоб уникнути втрат у якості під час компресії та появи артефактів, та тимчасово розміщується у ДТФ, щоб зменшити навантаження на програмний засіб;

Алгоритм поліпшення динамічного діапазону

Обробка та покращення широкого динамічного діапазону зображення.

Послідовність дія/відгук:

- а) результат, збережений тимчасово у результаті масштабування та обробки зображення завантажується у програму;
- б) перетворення колірної схеми допоможе при подальшій обробці HDR;
- в) після перетворення колірної схеми зображення копіюється на 2 екземпляри, після чого відбувається окрема обробка кожного із них;
- г) скопійовані зображення перетворюються у стан недоекспонованого та переекспонованого за допомогою виправлення гами та яскравості, що допоможе відтворити подібний до оригінального алгоритму результат;
- д) якщо обробка зображень пройшла успішно, тоді вони зберігаються у ДТФ;
- е) редаговані зображення та результат масштабування та поліпшення поєднуються за допомогою алгоритму об'єднання і злиття зображень та їх експозицій;
- ж) фінальний результат зберігається у теку результатів у форматі PNG, що допоможе зберегти оригінальну якість і перешкодить появі артефактів, які з'являються при використанні формату JPG.

3.2 Проектування програмного засобу

Розроблюваний програмний засіб буде працювати окремо та незалежно від інших програмних систем, що дозволить користувачу запускати його напряму, без використання інших програмних засобів.

Програмний засіб розробляється як монолітний застосунок, тобто матиме монолітну архітектуру. Метод, що розробляється буде містити у собі 3 менших методи, які функціонуватимуть як одне ціле, здійснюючи обробку зображення послідовно.

Програмний засіб можна поділити на такі компоненти:

- обробник;

- завантажувач;
- збільшувач;
- поліпшувач.

Обробник (handler) запускатиметься із програмним засобом. Він допоможе контролювати момент запуску/завершення роботи з програмним засобом, що допоможе при створенні системи тимчасових файлів.

Завантажувач (uploader) відповідає за завантаження зображень у програмний засіб. Він контролюватиме розширення та розміри зображення, що завантажуватиметься у програму.

Збільшувач (enlarger) та поліпшувач/покращувач (enhancer) здійснюватимуть масштабування зображення, його покращення та відновлення деталей, які могли втратитись під час масштабування.

Поліпшувач також можна поділити на 2 компоненти:

- поліпшення та відновлення зображення;
- покращення широкого динамічного діапазону;

На рисунку 3.1 показано діаграму компонентів програмного засобу.

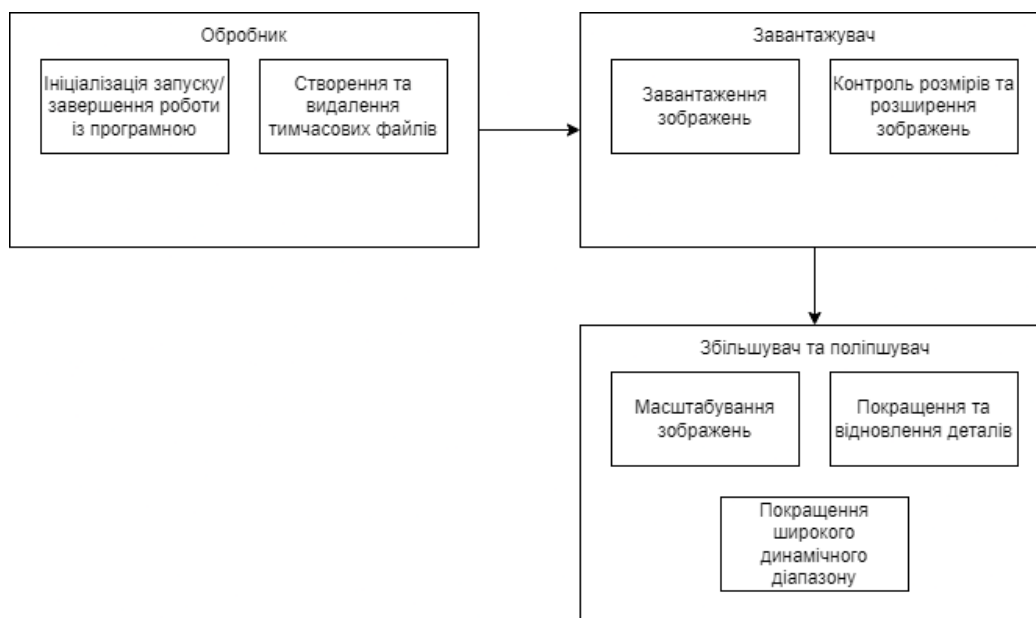


Рисунок 3.1 – Діаграма компонентів

Метод коригування та покращення широкого динамічного діапазону дозволить нормалізувати зображення та його колірний тон. Даний метод використовуватиме алгоритм злиття зображення Мертенса та експозиційний синтез зображень. Даний алгоритм використовуватиме 3 різні зображення із різними рівнями експозиції для брекети́нгу, тобто для злиття у одне зображення.

Даний алгоритм буде здійснювати обробку окремо, тобто в результаті користувач отримає 2 зображення, що покращить досвід роботи користувача із програмним засобом.

Для розробки методу покращення широкого динамічного діапазону використано злиття Мертенса. Порівнюючи із методом Дебевека та Робертсона (рисунок 3.2), метод злиття зображення Мертенса показує найбільш правильний результат, що використовувати дозволить після масштабування та покращення зображення та уникати спотворень кольорів.



Рисунок 3.2 – Порівняння алгоритмів злиття зображень

Для подальшої розробки методу покращення широкого динамічного діапазону (HDR) буде правильним рішенням розробити алгоритм у вигляді його схеми роботи. Це допоможе в подальшому орієнтуватись його розробці під час реалізації програмного засобу, що допоможе зекономити час та спершу зрозуміти структуру даного методу.

На рисунку 3.3 показано детальну схему роботи методу коригування та покращення широкого динамічного діапазону зображення (HDR).

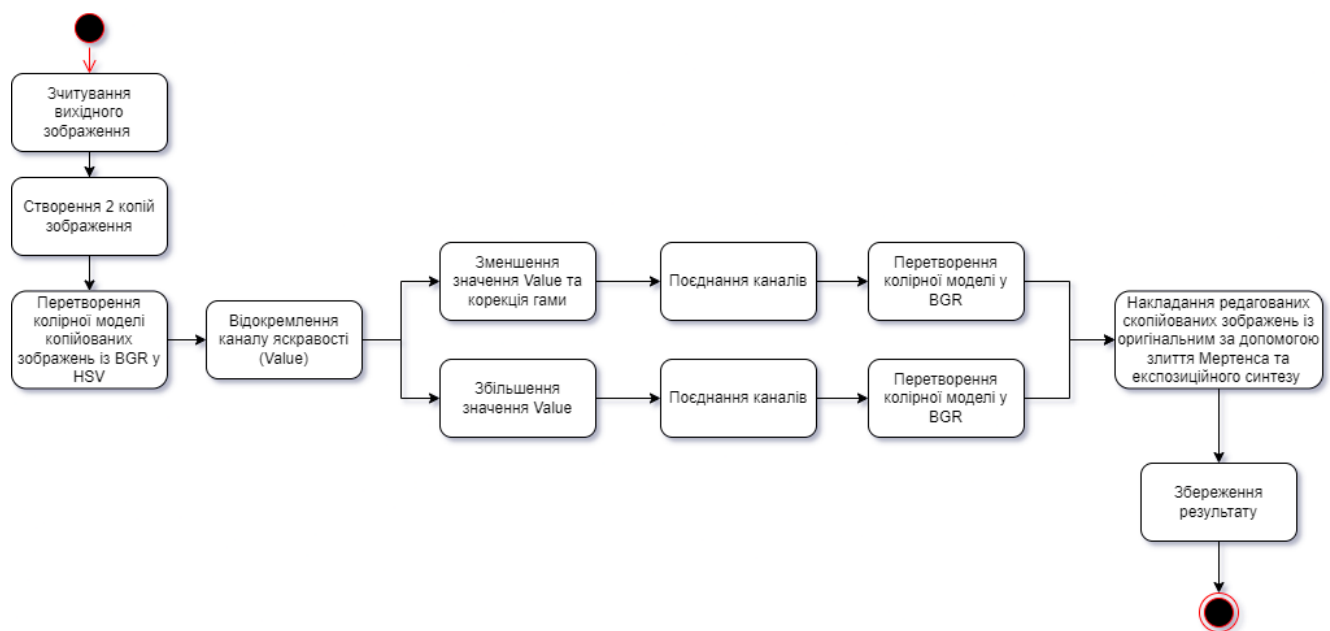


Рисунок 3.2 – Схема роботи методу HDR

На рисунку 3.3 показана діаграма послідовності, яка показує взаємодію компонентів програмного засобу, впорядкованих за часом. Вона допоможе зрозуміти послідовність дій та взаємодії між об'єктами програмного забезпечення, що розробляється.

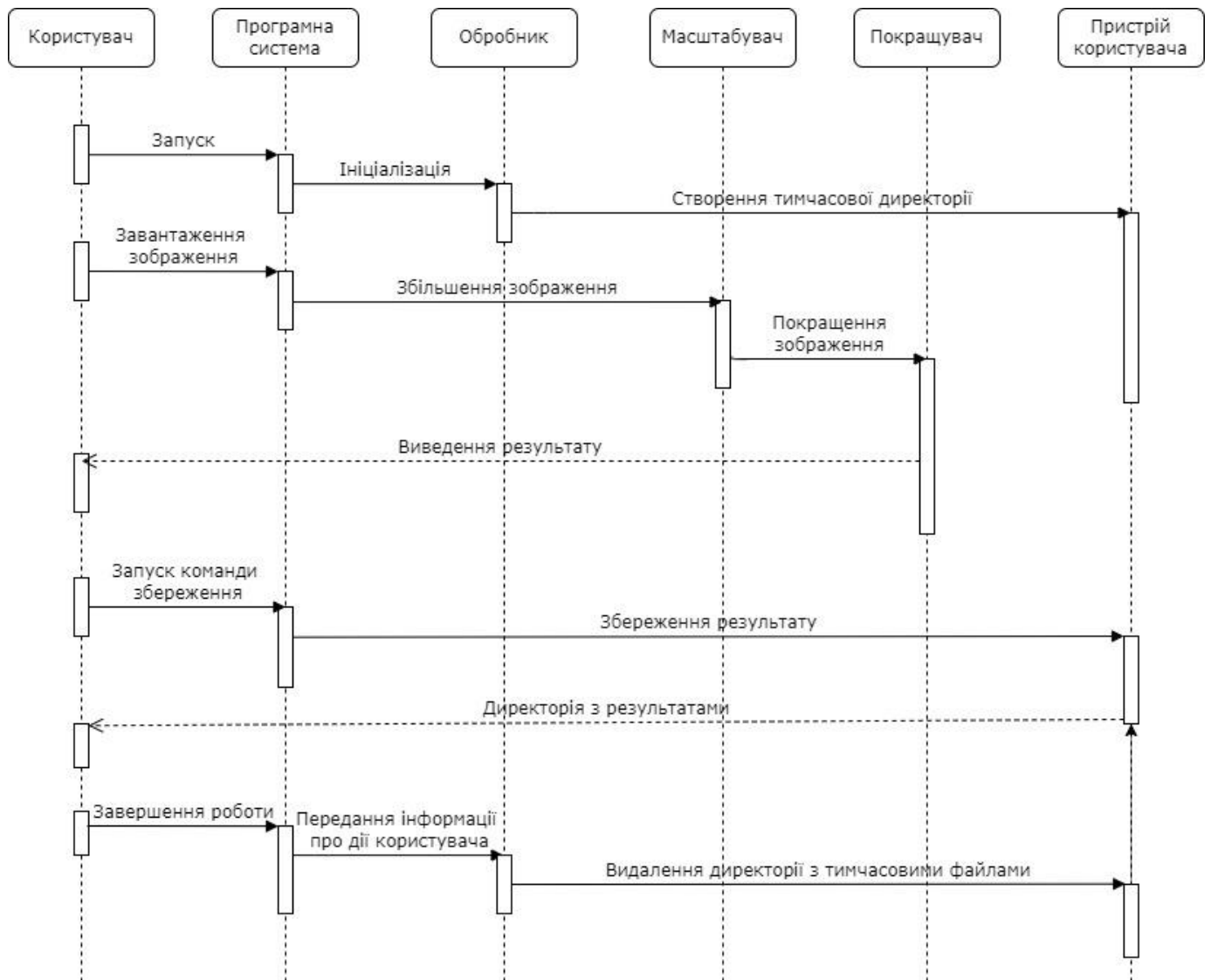


Рисунок 3.3 – Діаграма послідовності

3.3 Аналіз та вибір засобів програмної реалізації методу

При розробці програмного продукту варто врахувати наступні критерії:

- програмний засіб повинен бути простим у використанні та інтуїтивно зрозумілим користувачу;
- програмний засіб має бути незалежним від інших програмних продуктів, тобто працювати автономно;

- програмний продукт повинен обробляти зображення настільки швидко, наскільки це можливо;
- засіб повинен уникати затримок при обробці зображень;

З огляду на ці критерії було вирішено використовувати мову програмування Python, яка дає змогу використовувати безліч бібліотек та фреймворків, які допоможуть при подальшому масштабуванні та обробці цифрових зображень.

Одною із основних бібліотек, яка буде використовуватись для обробки зображень є бібліотека комп'ютерного зору OpenCV. Бібліотека має широкий функціонал для обробки і аналізу вмісту зображень та використовується для визначення об'єктів на зображенні, а також для перетворення зображень.

Також для роботи із растровими зображеннями доцільно використати бібліотеку Pillow (PIL), яка також використовується для обробки зображень. Для розробки програмного продукту буде використано модуль ImageTK, який дозволяє завантажувати зображення.

Для створення тимчасових файлів, які будуть потрібні для правильної роботи методу та програмного засобу загалом, буде використано модулі shutil та os. Дані модулі корисні при роботі із файлами чи колекціями файлів.

Для обробки бібліотек доцільно використовувати менеджер пакетів PIP, який дасть змогу керувати бібліотеками, що потрібні для правильного функціонування програмного забезпечення. PIP встановлює бібліотеки із репозиторію PyPi, який вміщує в собі понад сім мільйонів файлів.

Основним із критеріїв розробки програмного продукту є його простота у використанні. Тобто інтерфейс повинен бути максимально простим та інтуїтивно зрозумілим для користувача. Для цього буде використано стандартний пакет Tkinter, або інтерфейс Tk. Tk дозволяє створювати та керувати віджетами GUI та легко їх налаштовувати. Tk є новішим сімейством віджетів Tk та забезпечує більш сучасний вигляд для віджетів. Також він є частиною модулю Tk, починаючи із версії 8.5.

3.4 Висновки

У розділі було виконано аналіз вимог для розробки програмного засобу, а також визначено, що програмний засіб повинен уникати затримок та простоїв при масштабуванні та обробці зображень, а також витратити невелику кількість часу на роботу методу.

Було визначено вимоги до компонентів ПЗ, а також визначено основні компоненти потрібні для функціонування ПЗ. Основними компонентами програмного засобу є збільшувач та поліпшувач, за якими стоять метод масштабування та покращення зображення. Вони відіграють основну роль у роботі програмного засобу.

Важливими компонентами при роботі програмного засобу також є обробник та створення тимчасових файлів. Дані компоненти функціонуватимуть взаємопов'язано між собою, тобто обробник дозволить відстежувати дії користувача, а саме завершення роботи із програмним засобом, у результаті чого тимчасові файли, які були створенні при запуску програми, будуть видалятися, що дозволить уникнути накопичення «сміття», тобто непотрібних файлів. Також тимчасові файли матимуть атрибут «Прихований», що дозволить уникнути проблем із стабільністю роботи методу та програмного засобу, а також дозволить уникнути несанкціонованого доступу користувачем до тимчасових файлів, видалення яких може призвести до помилки програми.

Також додатково у розділі описано алгоритм роботи методу покращення широкого динамічного діапазону, що дозволить у подальшому нормалізувати та покращити кольоровий тон зображення.

У розділі також обрана архітектура ПЗ, визначено модулі ПЗ та описано призначення кожного модуля, здійснено їх декомпозицію на компоненти, побудовані діаграми UML (зокрема діаграми компонентів та послідовності).

Було обрано, а також обґрунтовано перелік технологій, які необхідні для вирішення задачі та реалізації методу та програмного засобу (Python, OpenCV, Tkinter та PIP).

Наступним етапом розробки програмного засобу буде реалізація спроектованого ПЗ, проведення його тестування, а також доведення ефективності розробленого методу масштабування зображень.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

4.1 Програмна реалізація

4.1.1 Структура та призначення модулів програми, їх взаємозв'язок

Програмний засіб складається з наступних компонентів:

Завантажувач – простий метод, який завантажуватиме вибране користувачем вхідне зображення з розширенням .PNG або .JPG, та виводитиме його на екран, а також інформацію про зображення, таку як шлях до файлу та його роздільна здатність.

Поліпшувач – прийматиме вхідне зображення та оброблятиме його за розробленим алгоритмом масштабування та покращення зображення. Додатково реалізовано алгоритм покращення динамічного діапазону зображення, завданням якого буде покращення уже готового результату та збільшення широкого динамічного діапазону цифрового зображення.

Обробник – дозволяє визначати, коли користувач запускає та закриває програмний засіб. Це допоможе реалізувати систему тимчасових файлів, що допоможе покращити обробку вхідного зображення та видаляти файли, які користувач вирішив не зберігати на пристрої. Тимчасові файли також відіграють важливу роль при обробці динамічного діапазону зображення.

Реалізація тимчасових файлів допоможе зменшити навантаження про програмний засіб під час обробки зображення та зберігати результати до моменту, коли користувач вирішить завершити роботу з програмним засобом і вирішить не зберігати результат.

Тимчасові файли матимуть атрибут FILE_ATTRIBUTE_HIDDEN, тобто будуть прихованими від звичайного користувача. Даний метод дозволить уникнути несанкціонованого доступу користувачем до «проміжних» файлів, які зберігають під час виконання програмного засобу.

4.1.2 Розробка програмних компонентів

Для початку було реалізовано алгоритм завантаження та зчитування зображення. Також реалізовано обмеження роздільної здатності зображення для уникнення перевантаження методу.

Фрагмент коду, що відповідає за зчитування коду показано у лістингу :

```
global img
global pathcv
try:
    root.filename = filedialog.askopenfilename(initialdir="/", title="Select file",
                                                filetypees=(("JPEG files", "*.jpg"),
("PNG files", "*.png"), ("all files", "*.*")))
    img = ImageTk.PhotoImage(Image.open(root.filename))
    prh = img.height()
    prw = img.width()
    if prh > 2048 or prw > 2048:
        tk.messagebox.showerror(title=None, message="Image is too big (height or
width are bigger than 2048)")
    else:
        pathh = root.filename
        pathcv = root.filename
        imag = cv2.imread(pathh, 1)
        cv2.imshow('Preview (' + str(prw) + 'x' + str(prh) + ')', imag)
        labelsize = Label(text="Image size: %dx%d" % (prw, prh), justify=LEFT)
        labelloc = Label(text="Image location: %s" % (root.filename), justify=LEFT)
        labelsize.place(relx=.1, rely=.7)
        labelloc.place(relx=.1, rely=.8)
```

Наступним кроком є реалізація методу масштабування та покращення растрового зображення.

Фрагмент програмного коду:

```
image = cv2.imread(pathcv)
scale_up = scaling.get()
res_inter_lin = cv2.resize(image, None, fx=scale_up, fy=scale_up,
interpolation=cv2.INTER_LINEAR_EXACT)
h_int, w_int, c_int = res_inter_lin.shape
bilateral_filter = cv2.bilateralFilter(src=res_inter_lin, d=9, sigmaColor=45,
sigmaSpace=45)
kernel = np.array([[ 0, -1, 0],
                   [-1, 5, -1],
                   [ 0, -1, 0]])
sharp = cv2.filter2D(src=bilateral_filter, ddepth=-1, kernel=kernel)
```

```

dst = cv2.detailEnhance(sharp, sigma_s=0.3, sigma_r=0.01)
clahe = cv2.createCLAHE(clipLimit=2., tileGridSize=(2, 2))
lab = cv2.cvtColor(dst, cv2.COLOR_BGR2LAB) # convert from BGR to LAB color space
l, a, b = cv2.split(lab) # split on 3 different channels
l2 = clahe.apply(l) # apply CLAHE to the L-channel
lab = cv2.merge((l2, a, b)) # merge channels
img2 = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
os.chdir(str(pathroot) + "\\temp")
cv2.imwrite("tempimg.png", img2)
cv2.imshow('RESULT (NON-HDR) (' + str(w_int) + 'x' + str(h_int) + ')', img2)

```

Для реєстрації виходу користувача із програмного продукту буде використано модуль `atexit`, який використовує реєстрацію та скасування реєстрації функцій. Зареєстровані функції автоматично виконуються після звичайного завершення роботи інтерпретатора, а модуль `atexit` дозволяє запускати ці функції у зворотному порядку, у якому вони були зареєстровані.

Для реалізації тимчасових файлів використано модуль `ctypes`, який дозволяє функції в DLL або у спільних бібліотеках. У даному випадку створеній теці присвоєно атрибут `FILE_ATTRIBUTE_HIDDEN` використовуючи `kernel32.dll`.

У наступному лістингу показано фрагмент коду, який відповідає за створення директорії тимчасових файлів та за обробку дій користувача, а саме видалення тимчасових файлів.

```

pathroot = Path().absolute()
if not os.path.exists(str(pathroot) + "\\temp"):
    os.mkdir("temp")
tempresfolder = (str(pathroot) + "\\temp")
FILE_ATTRIBUTE_HIDDEN = 0x02
ret = ctypes.windll.kernel32.SetFileAttributesW(tempresfolder, FILE_ATTRIBUTE_HIDDEN)

def exit_handler():
    shutil.rmtree(tempresfolder)

atexit.register(exit_handler)

```

Додатково було реалізовано алгоритм покращення широкого динамічного діапазону. Даний алгоритм буде використовувати злиття Мертенса та експозиційного синтезу. В обробці цифрових зображень та у комп'ютерній графіці експозиційним

синтезом або злиттям експозиції називають техніку змішування кількох експозицій однієї сцени (брекетинг) в одне зображення.

У даному випадку алгоритм дій буде відрізнятися від традиційного методу покращення широкого динамічного діапазону, який використовується безпосередньо при зйомці зображення.

Даний алгоритм буде використовувати колірний простір HSV, тобто Hue – відтінок, Saturation – насиченість та Value – значення яскравості. Цей колірний простір описує кольори (відтінок) з точки зору їх відтінку (насиченості або кількості сірого) і значення яскравості.

Для розробки методу покращення широкого динамічного діапазону буде використано канал значення яскравості у поєднанні з корекцією гама. Значення яскравості працює в поєднанні з насиченістю й описує яскравість або інтенсивність кольору від 0 до 100 відсотків, де 0 означає повністю чорний колір, а 100 — найяскравіший і найбільш яскравий колір.

Код методу покращення широкого динамічного діапазону показано у лістингу:

```
prehdr = cv2.imread("tempimg.png")
hsv = cv2.cvtColor(prehdr, cv2.COLOR_BGR2HSV)
hsv1 = hsv.copy()

h, s, v = cv2.split(hsv)
value1 = 75
lim = value1
v[v < lim] = 0
v[v >= lim] -= value1
mid = 0.3
mean = np.mean(v)
gamma = math.log(mid * 255) / math.log(mean)
val_gamma = np.power(v, gamma).clip(0, 255).astype(np.uint8)
low1 = cv2.merge((h, s, val_gamma))

h1, s1, v1 = cv2.split(hsv1)
value2 = 50
lim1 = 255 - value2
v1[v1 > lim1] = 255
v1[v1 <= lim1] += value2
high1 = cv2.merge((h1, s1, v1))

lowex = cv2.cvtColor(low1, cv2.COLOR_HSV2BGR)
highex = cv2.cvtColor(high1, cv2.COLOR_HSV2BGR)
cv2.imwrite("low.png", lowex)
```

```

cv2.imwrite("high.png", highex)

img_fn = ["low.png",
         "tempimg.png",
         "high.png"]

images = []
for filename in img_fn:
    im = cv2.imread(filename)
    images.append(im)
alignMTB = cv2.createAlignMTB()
alignMTB.process(images, images)
mergeMertens = cv2.createMergeMertens()
exposureFusion = mergeMertens.process(images)
cv2.imwrite("temphdr.png", exposureFusion * 255)
hdrimgpath = str(tempresfolder) + "\\temphdr.png"
hdrimg = cv2.imread(hdrimgpath)
cv2.imshow('RESULT (HDR) (' + str(w_int) + 'x' + str(h_int) + ')', hdrimg)

```

4.2 Результати тестування програмного засобу та їх аналіз

4.2.1 Вибір методу тестування

Після проектування методу та програмного засобу масштабування зображення необхідно провести їх тестування, а також переконатись, що метод та програмний засіб є повністю функціональними та задовольняють усі умови.

У даному випадку буде дуже доречним провести порівняння із оригінальним зображення, визначити певні параметри та значення їх схожості між собою, що покаже дієвість методу, а також провести тестування, яке дозволить визначити, скільки часу потрібно для обробки та виконання методів.

Для порівняння результатів буде використано метрики для визначення структурної подібності двох зображень. Однією з таких метрик є SSIM.

Індекс структурної подібності (structural similarity index measure або коротко SSIM) – це метод прогнозування сприйнятої якості цифрових телевізійних та кінематографічних зображень, а також інших типів цифрових зображень і відео. SSIM використовується для вимірювання подібності між двома зображеннями [16].

SSIM – це модель на основі сприйняття, яка розглядає деградацію зображення як сприйнятту зміну структурної інформації та включає важливі явища, а також включає маскування контрасту та яскравості. Основна різниця із іншими методами, наприклад із MSE чи PSNR, полягає в тому, що використання даних методів дозволить оцінити лише абсолютні похибки.

Маскування яскравості – це певне явище, під час якого спотворення зображення (у цьому контексті) мають тенденцію бути менш помітними в яскравих областях, тоді як при маскуванні контрасту спотворення стають менш помітними при наявності значної активності чи «текстури» на зображенні.

Також існує більш сучасніша форма SSIM, яку називають багатомасштабною SSIM (Multiscale SSIM або MS-SSIM). Вона здійснюється у кількох масштабах з використанням багатетапного процесу зменшення дискретизації [17].

Також для визначення якості зображення та порівняння його із іншим зображення використовують метрику точності візуальної інформації VIF (Visual Information Fidelity), або його ще називають індексом Шейха-Бовіка. Точність візуальної інформації дозволяє розрахувати індекс оцінки якості цифрового зображення. Цей індекс був заснований використовуючи статистику природної сцени та поняття інформації про цифрове зображення, яка була отримана завдяки зоровій системі людини.

Індекс VIF використовує статистичні моделі природної сцени (NSS) поєднуючи їх з моделлю спотворень (каналів) для кількісної оцінки інформації, яка передається поміж контрольними та тестовими зображеннями [18].

Також окрім вище згаданих метрик буде доречно використати універсальний індекс якості зображення (UQI). UQI допомагає оцінити якість зображення використовуючи втрати кореляції, спотворення яскравості та спотворення контрасту.

4.2 Розробка тестових сценаріїв

Метою модульного тестування є дослідження програмної системи на її відповідність поставленим вимогам, які були наведені у специфікації вимог. У таблиці 4.1 було наведено перелік тестових випадків та було описано очікуваний результат роботи кожної функції програмного засобу.

Таблиця 4.1 – Тестові випадки

Ч.ч	Функція	Вихідні дані	Очікуваний результат
1	2	3	4
2	Завантажувач	Вибране користувачем зображення	Зчитування та показ вибраного користувачем зображення
3	Збільшувач та поліпшувач	Масштабоване зображення	Масштабує та покращує зображення після цього покращує його широкий динамічний діапазон
4	Обробник	Результат виконання операції	Створення та видалення тимчасових файлів при початку та завершенні роботи із програмою
5	Збереження результату	Збережене масштабове зображення та HDR-зображення	Зберігає результати роботи програмного забезпечення

Використання метрик схожості зображення допоможе довести достовірність результатів роботи методу та програмного засобу загалом.

Було проведено по 20 тестів з використанням різних параметрів, а саме коефіцієнт збільшення. Це допомогло порівняти середній час виконання методів та порівняти його.

4.3 Аналіз результатів тестування

Використання метрик для порівняння схожості та якості зображень є важливим фактором у розробці методу та програмного засобу масштабування зображень.

Проведення тестування методу підтвердить його працездатність та достовірність результатів.

Доречним буде порівняння оригінального зображення, звичайного результату та результату із покращеним динамічним діапазоном. Для цього буде використано метрики SSIM, MS-SSIM, VIF та UQI. Метрики схожості зображення потребують зображень однакових розмірів, тому оригінальне зображення буде збільшено до розмірів результатів за допомогою звичайного збільшення роздільної здатності.

На рисунку 4.1 показано результати проведеного порівняння схожості зображень між собою за допомогою модулю `sewar`.

```
MS-SSIM score (ORIGINAL - NON-HDR): (0.909119213901273+0j)
SSIM score (ORIGINAL - NON-HDR): (0.8073058863496456, 0.8268789134550741)
VIF Index (ORIGINAL - NON-HDR): 0.32856901290918433
UQI score (ORIGINAL - NON-HDR): 0.9318878986423188
MS-SSIM score (ORIGINAL - HDR): (0.9167504595547992+0j)
SSIM score (ORIGINAL - HDR): (0.7947657483389324, 0.8252543813304274)
VIF score (ORIGINAL - HDR): 0.31467733647072893
UQI score (ORIGINAL - HDR): 0.907468344967143
MS-SSIM score (NON-HDR - HDR): (0.9904896563356189+0j)
SSIM score (NON-HDR - HDR): (0.9727385412827158, 0.9848712333550572)
VIF Index (NON-HDR - HDR): 0.9999999999877606
UQI score (NON-HDR - HDR): 0.9693191841644326
```

Рисунок 4.1 – Результати метрик схожості зображень

Результати метрики схожості зображення показують, що даний метод масштабування зображення є повністю працездатним та задовольняє усі умови.

Наступним важливим кроком є проведення тестування часу роботи методу. Для більш детальних результатів було проведено окремо тестування масштабування зображення та тестування покращення широкого динамічного діапазону зображення.

Для проведення тестування було використано зображення із роздільною здатністю 640 на 399 пікселів та виконано його масштабування у 2, 2,5 та у 3 рази.

Для отримання більш детальних результатів було проведено 20 тестів для кожного із методів та було знайдено середній час роботи цих методів, що допоможе більш детально з'ясувати його працездатність та швидкодію.

На рисунках 4.2 – 4.4 показано результати часового тестування працездатності методу масштабування зображення.

```
===== EXECUTION TEST (2x scale)=====
EXECUTION # 1 : 0.8593809604644775 seconds
EXECUTION # 2 : 0.6249933242797852 seconds
EXECUTION # 3 : 0.5312509536743164 seconds
EXECUTION # 4 : 0.5781238079071045 seconds
EXECUTION # 5 : 1.8749985694885254 seconds
EXECUTION # 6 : 0.515625 seconds
EXECUTION # 7 : 0.4843757152557373 seconds
EXECUTION # 8 : 0.4843757152557373 seconds
EXECUTION # 9 : 0.5312566757202148 seconds
EXECUTION # 10 : 0.5160295963287354 seconds
EXECUTION # 11 : 0.5156242847442627 seconds
EXECUTION # 12 : 0.48441195487976074 seconds
EXECUTION # 13 : 0.5004072189331055 seconds
EXECUTION # 14 : 0.5312502384185791 seconds
EXECUTION # 15 : 0.4999990463256836 seconds
EXECUTION # 16 : 0.5160353183746338 seconds
EXECUTION # 17 : 0.546872615814209 seconds
EXECUTION # 18 : 0.4843761920928955 seconds
EXECUTION # 19 : 0.46916699409484863 seconds
EXECUTION # 20 : 0.4843754768371582 seconds
AVERAGE EXECUTION TIME: 0.5729966504233224 seconds
```

Рисунок 4.2 – Результати роботи методу масштабування при коефіцієнті 2.0

```
===== EXECUTION TEST (2.5x scale)=====
EXECUTION # 1 : 0.9725058078765869 seconds
EXECUTION # 2 : 0.7690808773040771 seconds
EXECUTION # 3 : 0.7060341835021973 seconds
EXECUTION # 4 : 0.7368237972259521 seconds
EXECUTION # 5 : 1.6158056259155273 seconds
EXECUTION # 6 : 1.5701749324798584 seconds
EXECUTION # 7 : 0.7213685512542725 seconds
EXECUTION # 8 : 0.70627760887146 seconds
EXECUTION # 9 : 0.7062516212463379 seconds
EXECUTION # 10 : 0.7064023017883301 seconds
EXECUTION # 11 : 0.7302718162536621 seconds
EXECUTION # 12 : 0.7130746841430664 seconds
EXECUTION # 13 : 0.7369091510772705 seconds
EXECUTION # 14 : 0.7546653747558594 seconds
EXECUTION # 15 : 0.7827920913696289 seconds
EXECUTION # 16 : 0.7377073764801025 seconds
EXECUTION # 17 : 0.7531826496124268 seconds
EXECUTION # 18 : 0.7842943668365479 seconds
EXECUTION # 19 : 0.70615553855896 seconds
EXECUTION # 20 : 0.7218976020812988 seconds
AVERAGE EXECUTION TIME: 0.7919845694587344 seconds
```

Рисунок 4.3 – Результати роботи методу масштабування при коефіцієнті 2.5

```
===== EXECUTION TEST (3x scale)=====
EXECUTION # 1 : 1.3180420398712158 seconds
EXECUTION # 2 : 2.4467861652374268 seconds
EXECUTION # 3 : 1.318394660949707 seconds
EXECUTION # 4 : 1.051589012145996 seconds
EXECUTION # 5 : 1.0351614952087402 seconds
EXECUTION # 6 : 1.0199880599975586 seconds
EXECUTION # 7 : 1.051527738571167 seconds
EXECUTION # 8 : 1.0505735874176025 seconds
EXECUTION # 9 : 1.0365192890167236 seconds
EXECUTION # 10 : 1.019984483718872 seconds
EXECUTION # 11 : 1.0203886032104492 seconds
EXECUTION # 12 : 1.0827889442443848 seconds
EXECUTION # 13 : 1.0668306350708008 seconds
EXECUTION # 14 : 1.0201921463012695 seconds
EXECUTION # 15 : 1.0200386047363281 seconds
EXECUTION # 16 : 1.034954309463501 seconds
EXECUTION # 17 : 1.0516002178192139 seconds
EXECUTION # 18 : 1.0195610523223877 seconds
EXECUTION # 19 : 1.0205872058868408 seconds
EXECUTION # 20 : 1.0205330848693848 seconds
AVERAGE EXECUTION TIME: 1.0812400636218844 seconds
```

Рисунок 4.4 – Результати роботи методу масштабування при коефіцієнті 3.0

На рисунках 4.5 – 4.7 показано результати роботи методу покращення широкого динамічного діапазону зображення.

```
===== FAKE-HDR EXECUTION TEST (2x scale)=====
EXECUTION # 1 : 0.4552726745605469 seconds
EXECUTION # 2 : 0.47124290466308594 seconds
EXECUTION # 3 : 0.48662471771240234 seconds
EXECUTION # 4 : 0.45476293563842773 seconds
EXECUTION # 5 : 0.4674527645111084 seconds
EXECUTION # 6 : 0.47111988067626953 seconds
EXECUTION # 7 : 0.45525455474853516 seconds
EXECUTION # 8 : 0.45535945892333984 seconds
EXECUTION # 9 : 0.4554150104522705 seconds
EXECUTION # 10 : 0.4554252624511719 seconds
EXECUTION # 11 : 0.4552290439605713 seconds
EXECUTION # 12 : 0.45528340339660645 seconds
EXECUTION # 13 : 0.45530104637145996 seconds
EXECUTION # 14 : 0.48663854598999023 seconds
EXECUTION # 15 : 0.4706387519836426 seconds
EXECUTION # 16 : 0.45650386810302734 seconds
EXECUTION # 17 : 0.45784735679626465 seconds
EXECUTION # 18 : 0.4590179920196533 seconds
EXECUTION # 19 : 0.4600250720977783 seconds
EXECUTION # 20 : 0.45447659492492676 seconds
AVERAGE EXECUTION TIME: 0.46194459199905397 seconds
```

Рисунок 4.5 – Результати Fake-HDR при коефіцієнті 2.0

```
===== FAKE-HDR EXECUTION TEST (2.5x scale)=====
EXECUTION # 1 : 0.7062685489654541 seconds
EXECUTION # 2 : 0.7375528812408447 seconds
EXECUTION # 3 : 0.7056629657745361 seconds
EXECUTION # 4 : 0.7202582359313965 seconds
EXECUTION # 5 : 0.7062716484069824 seconds
EXECUTION # 6 : 0.7060270309448242 seconds
EXECUTION # 7 : 0.7224388122558594 seconds
EXECUTION # 8 : 0.6906547546386719 seconds
EXECUTION # 9 : 0.7061631679534912 seconds
EXECUTION # 10 : 0.6901261806488037 seconds
EXECUTION # 11 : 0.7062208652496338 seconds
EXECUTION # 12 : 0.706312894821167 seconds
EXECUTION # 13 : 0.7529997825622559 seconds
EXECUTION # 14 : 0.7556455135345459 seconds
EXECUTION # 15 : 0.927903413772583 seconds
EXECUTION # 16 : 0.7686588764190674 seconds
EXECUTION # 17 : 0.7692255973815918 seconds
EXECUTION # 18 : 0.8196425437927246 seconds
EXECUTION # 19 : 0.925187349319458 seconds
EXECUTION # 20 : 0.8161051273345947 seconds
AVERAGE EXECUTION TIME: 0.7519663095474243 seconds
```

Рисунок 4.5 – Результати Fake-HDR при коефіцієнті 2.5

```
===== FAKE-HDR EXECUTION TEST (3x scale)=====
EXECUTION # 1 : 1.0353193283081055 seconds
EXECUTION # 2 : 1.1301257610321045 seconds
EXECUTION # 3 : 1.0043952465057373 seconds
EXECUTION # 4 : 0.9890096187591553 seconds
EXECUTION # 5 : 0.9890499114990234 seconds
EXECUTION # 6 : 1.0357677936553955 seconds
EXECUTION # 7 : 0.9891793727874756 seconds
EXECUTION # 8 : 0.9888291358947754 seconds
EXECUTION # 9 : 1.0354437828063965 seconds
EXECUTION # 10 : 1.004621982574463 seconds
EXECUTION # 11 : 1.0039353370666504 seconds
EXECUTION # 12 : 1.0987026691436768 seconds
EXECUTION # 13 : 0.988461971282959 seconds
EXECUTION # 14 : 0.9887604713439941 seconds
EXECUTION # 15 : 1.301347017288208 seconds
EXECUTION # 16 : 1.0042476654052734 seconds
EXECUTION # 17 : 1.0828564167022705 seconds
EXECUTION # 18 : 1.0048413276672363 seconds
EXECUTION # 19 : 1.0201008319854736 seconds
EXECUTION # 20 : 1.5528788566589355 seconds
AVERAGE EXECUTION TIME: 1.0623937249183655 seconds
```

Рисунок 4.5 – Результати Fake-HDR при коефіцієнті 3.0

Як показують результати, на виконання методу програмне забезпечення витрачає від 1 до 2,2 секунд, в залежності від коефіцієнту масштабування, при цьому середня різниця у часі при збільшені коефіцієнту масштабування дорівнює приблизно 0,25-0,3 секунди для кожного з методів або 0,5-0,7 секунд у загальному.

Останнім тестуванням буде візуальне порівняння зображень та порівняння їх гістограм освітленості. Це допоможе візуально порівняти різницю між оригінальним зображенням та результатами.

На рисунку 4.6 показано візуальне порівняння результатів розробленого методу із оригінальним зображенням.



Рисунок 4.6 – Візуальне відображення результатів

Діаграма освітленості дозволить розглянути зміни у експозиції та яскравості зображення. На рисунку 4.7 зображено порівняння гістограм освітленості, де червоним кольором показана гістограма оригінального зображення, синім – гістограма масштабованого зображення та зеленим – гістограма HDR-зображення.

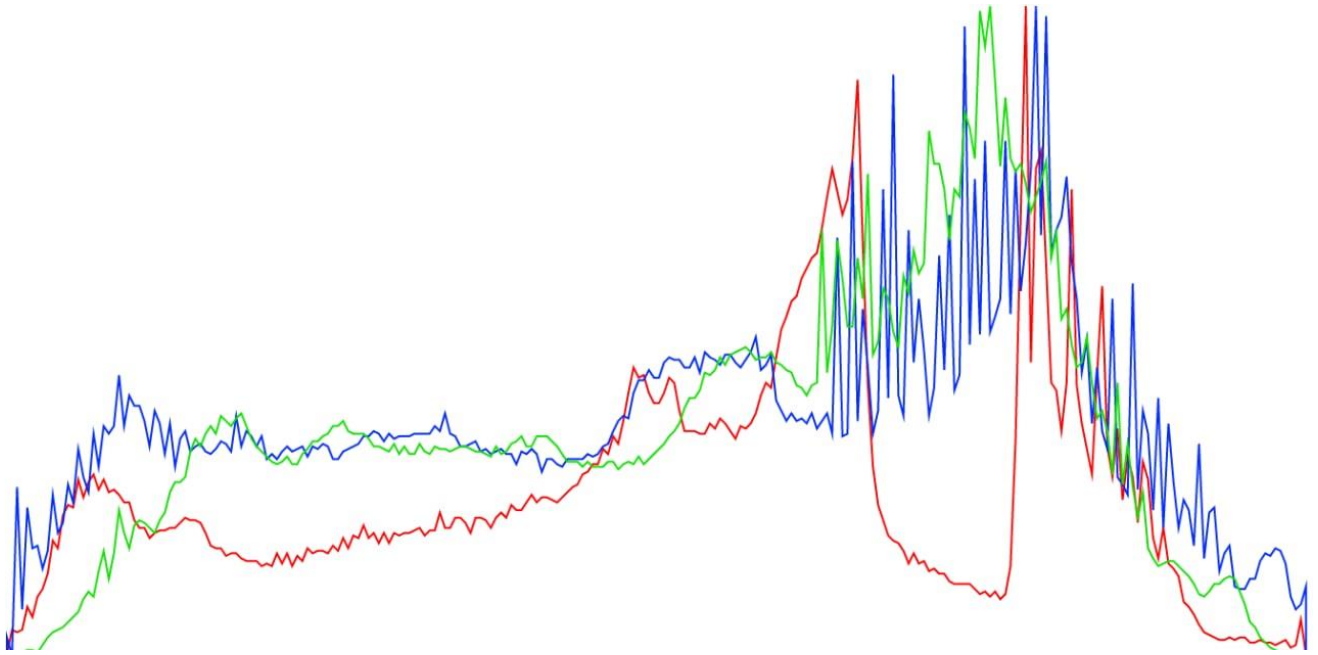


Рисунок 4.7 – Порівняння гістограм освітленості

У результаті порівняння гістограм освітленості помітно, що даний метод дозволив покращити середні тони зображення та усунути «діри» у тонах, якщо вони наявні, що допоможе нормалізувати зображення.

Також доречним буде проведення порівняння результатів роботи методу із наявними програмними рішеннями. Це допоможе провести аналіз та довести достовірність результатів розробленого методу.

Для цього було вибрано нейронні мережі Real-ESRGAN та ESDR, а також результат масштабування та покращення зображення за допомогою розробленого методу та з використанням розробленого методу Fake-HDR, тобто результат із покращеними широким динамічним діапазоном.

На рисунку 4.8 показано порівняння результатів роботи різних методів масштабування зображення. Як показують ці результати, розроблений метод дозволяє зберегти велику кількість деталей зображення, які втрачаються при використанні згорткових мереж.



Рисунок 4.8 – Порівняння результатів роботи методів масштабування

4.4 Інтеграція та налаштування програмного засобу

Програмне забезпечення встановлюватиметься та функціонуватиме незалежно та автономно від інших програмних засобів. Програмний засіб можна буде встановлювати та використовувати лише на персональному комп'ютері при наявності операційної системи Windows.

Для функціонування програмного продукту потрібно встановити Python 3. Для уникнення помилок та проблем із запуском чи встановленням рекомендується використовувати Python версії 3.8 та новіші.

Необхідні бібліотеки та модулі будуть встановлені автоматично під час установки програмного засобу.

Програмний продукт запускатиметься за допомогою виконавчого файлу.

4.5 Висновки

У даному розділі було описано модулі та компоненти програмного продукту, його складові та їх взаємодія.

Було здійснено програмну реалізацію методу та програмного засобу, описано деталі функціонування його компонентів та модулів.

Також було розглянуто реалізацію поліпшення широкого динамічного діапазону зображення, що допомогло при його нормалізації.

Було проведено детальне тестування методу та програмного засобу з використання метрик схожості цифрових зображень, а також було проведено тестування швидкодії методів для підтвердження його працездатності та виконання умов та критерій, які були поставлені для його розробки.

Як показує порівняння результатів роботи різних методів, розроблений метод показує гарний результат і при порівнянні дрібних деталей він показує один із кращих результатів, що доводить їх достовірність.

В результаті тестування з'ясовано, що різниця у роботі методу при використанні різних коефіцієнтів масштабування дорівнює 0,5-0.7 секунд.

ВИСНОВКИ

У процесі дипломного проектування досліджено галузь масштабування та сучасні методи і засоби масштабування зображень, було визначено невирішені проблеми у галузі, на основі чого розроблено власний метод масштабування зображення, виконано його програмну реалізацію та тестування.

У першому розділі дипломної роботи було досліджено та проаналізовано предметну область масштабування зображень, публікації та джерела, існуючі методи та засоби масштабування зображень для виявлення основних невирішених питань та проблем. На основі проведеного аналізу було визначено основні методологічні підходи до вирішення задачі масштабування зображення, а також виконана розгорнута постановка задач дослідження.

У другому розділі було здійснено аналіз моделей та методів масштабування зображення. Деякі невирішені проблеми запропоновано вирішити завдяки інтерполяції зображення із використанням подальшої обробки зображень. Було розглянуто основні проблеми та переваги основних алгоритмів інтерполяції, розглянуто ядра інтерполяції, що дозволило більш детально оглянути структуру та функціонал кожного із алгоритмів, завдяки чому було вибрано один із них, який використовуватиметься при розробці власного методу.

У третьому розділі було описано технології реалізації методу масштабування зображень, а саме було здійснено аналіз вимог до ПЗ та розробку його структури. Також було обґрунтовано вибір засобів для реалізації програмного засобу. Для програмної реалізації методу використано технології OpenCV, Pillow, Tkinter та інші, а для тестування – метрики порівняння схожості двох зображень.

У четвертому розділі було проведено програмну реалізацію та тестування програмної системи. У процесі проведення тестування було з'ясовано, що

розроблений метод не додає спотворень зображенням, а також є ефективним при використанні на недостатньо потужних пристроях.

Розроблений метод був реалізований як монолітний програмний застосунок, що працюватиме автономно і не потребуватиме встановлення додаткових програмних засобів, а також повністю не буде залежати від інших програмних засобів.

Практична апробація результатів свідчить про адекватність методу, його ефективність у порівнянні із уже існуючими програмними рішеннями, а також доводить працездатність та функціональну придатність програмного засобу, що був розроблений на основі цього методу. Результати показують, що час роботи методу при різних сценаріях використання не відрізняється суттєво по часу виконання. Проведено порівняння результатів роботи методу використовуючи метрики схожості зображень. Проведене тестування свідчить про відсутність спотворень під час виконання методу.

Розроблений метод можна рекомендувати до використання компаніям, які займаються обробкою зображень та графічним дизайнерам.

Таким чином, результати дослідження за темою дипломної роботи мають наукову новизну та практичну цінність.

Результати дослідження опубліковані у збірнику наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022».

Копія наукової публікації подана у додатку Б.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gonzalez R. C. Digital image processing. 2nd ed. Englewood Cliffs. [Electronic resource] / R. C. Gonzalez, R. E. Woods. – 2002. – Available: https://www.academia.edu/25992484/Digital_Image_Processing_Second_Edition.
2. Katsaggelos A.K. Digital image restoration. [Electronic resource] / A. K. Katsaggelos, M. R. Banham – Available: <https://ieeexplore.ieee.org/abstract/document/581363>.
3. Katsaggelos A.K. Super resolution of images and video. [Electronic resource] / A. K. Katsaggelos, Rafael Molina, Javier Mateos – Available: <https://books.google.com.ua/books?id=MY5eAQAAQBAJ&lpg=PR1&pg=PR1#v=onepage&q&f=false>.
4. J. C. Brailean. Noise Reduction Filters for Dynamic Image Sequences: A Review. [Electronic resource] / J. C. Brailean, R. P. Kleihorst, S. N. Efstratiadis, A. K. Katsaggelos, R.L. Legendijk – Available: https://www.researchgate.net/publication/2984937_Noise_Reduction_Filters_for_Dynamic_Image_Sequences_A_Review.
5. Christian Ledig. Photo-realistic single image super-resolution using a generative adversarial network. [Electronic resource] / Christian Ledig, Lucas Theis, Ferenc Husz'ar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang – Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.pdf.
6. Overview of GAN Structure. *Google Developers*. URL: https://developers.google.com/machine-learning/gan/gan_structure
7. Xintao Wang. Real-ESRGAN: Training Real-World Blind Super-Resolution With Pure Synthetic Data. [Electronic resource] / Xintao Wang, Liangbin Xie, Chao Dong, Ying

Shan – Available:

https://openaccess.thecvf.com/content/ICCV2021W/AIM/papers/Wang_Real-ESRGAN_Training_Real-World_Blind_Super-Resolution_With_Pure_Synthetic_Data_ICCVW_2021_paper.pdf.

8. Tero Karras. Analyzing and Improving the Image Quality of StyleGAN. [Electronic resource] / Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila – Available: <https://arxiv.org/abs/1912.04958>.

9. Adelson EH, Anderson CH, Bergen JR, Burt PJ, Ogden JM. Pyramid methods in image processing. [Electronic resource] – Available: http://persci.mit.edu/pub_pdfs/RCA84.pdf.

10. Wei-Sheng Lai. Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks. [Electronic resource] / Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, Ming-Hsuan Yang – Available: <https://arxiv.org/pdf/1710.01992.pdf>.

11. Графічний редактор Adobe Photoshop. Adobe Photoshop: URL: <https://www.adobe.com/>.

12. Bilinear interpolation. *X-Engineer.org*. URL: <https://x-engineer.org/bilinear-interpolation/>

13. P. Getreuer. Linear Methods for Image Interpolation. [Electronic resource]. / Pascal Getreuer – Режим доступу: https://www.ipol.im/pub/art/2011/g_lmii/?utm_source=doi.

14. Image Kernels. *Medium.com*. URL: <https://medium.com/practical-data-science-and-engineering/image-kernels-88162cb6585d>.

15. CLAHE Histogram Equalization – OpenCV. URL: <https://www.geeksforgeeks.org/clahe-histogram-equalization-opencv/>

16. Zhou Wang. Multi-scale structural similarity for image quality assessment. [Electronic resource] / Zhou Wang, Eero P. Simoncelli, Alan C. Bovik – Available: <https://www.cns.nyu.edu/pub/eero/wang03b.pdf>.

17. SSIM/MS-SSIM. *Vicuesoft.com*. URL: <https://vicuesoft.com/glossary/term/ssim-ms-ssim/>

18. H.R. Sheikh. An information fidelity criterion for image quality assessment using natural scene statistics / Sheikh H.R., Bovik A.C., G. de Veciana // *IEEE Transactions on Image Processing*. – Dec. 2005. – vol.14, no.12, pp. 2117- 2128.

ДОДАТОК А
(обов'язковий)

ПРОГРАМНИЙ КОД

```
import math
import shutil
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
from tkinter import *
import PIL
from PIL import Image, ImageTk
import numpy as np
import cv2
import os
import ctypes
from pathlib import Path
import atexit
import subprocess
from datetime import datetime

root = tk.Tk()
root.title('ENLIMG')
root.geometry('400x200')
root.resizable(0, 0)
scale = 1

scaling = DoubleVar()

sc1 = Radiobutton(root, text="2x", variable=scaling, value=2.0)
sc2 = Radiobutton(root, text="2.5x", variable=scaling, value=2.5)
sc3 = Radiobutton(root, text="3x", variable=scaling, value=3.0)

pathroot = Path().absolute()
if not os.path.exists(str(pathroot) + "\\temp"):
    os.mkdir("temp")
tempresfolder = (str(pathroot) + "\\temp")
FILE_ATTRIBUTE_HIDDEN = 0x02
ret = ctypes.windll.kernel32.SetFileAttributesW(tempresfolder, FILE_ATTRIBUTE_HIDDEN)

def exit_handler():
    shutil.rmtree(tempresfolder)

atexit.register(exit_handler)

def uploading():
    global img
    global pathcv
    try:
        root.filename = filedialog.askopenfilename(initialdir="/", title="Select
file",
```

```

filetypes=(("JPEG files",
".*.jpg"), ("PNG files", "*.png"), ("all files", "*.*"))
img = ImageTk.PhotoImage(Image.open(root.filename))
prh = img.height()
prw = img.width()
if prh > 2048 or prw > 2048:
    tk.messagebox.showerror(title=None, message="Image is too big (height or
width are bigger than 2048)")
else:
    pathh = root.filename
    pathcv = root.filename
    imag = cv2.imread(pathh, 1)
    cv2.imshow('Preview (' + str(prw) + 'x' + str(prh) + ')', imag)
    labelsize = Label(text="Image size: %dx%d" % (prw, prh), justify=LEFT)
    labelloc = Label(text="Image location: %s" % (root.filename),
justify=LEFT)
    labelsize.place(relx=.1, rely=.7)
    labelloc.place(relx=.1, rely=.8)
    enlargebt["state"] = "enable"
    sc1["state"] = "active"
    sc2["state"] = "active"
    sc3["state"] = "active"

except PIL.UnidentifiedImageError:
    tk.messagebox.showerror(title=None, message="Please choose image (.jpg or
.png)")

def enlarge():
    try:
        image = cv2.imread(pathcv)
        scale_up = scaling.get()
        res_inter_lin = cv2.resize(image, None, fx=scale_up, fy=scale_up,
interpolation=cv2.INTER_LINEAR_EXACT)
        h_int, w_int, c_int = res_inter_lin.shape
        bilateral_filter = cv2.bilateralFilter(src=res_inter_lin, d=9, sigmaColor=45,
sigmaSpace=45)
        kernel = np.array([[ 0, -1, 0],
                           [-1, 5, -1],
                           [ 0, -1, 0]])
        sharp = cv2.filter2D(src=bilateral_filter, ddepth=-1, kernel=kernel)
        dst = cv2.detailEnhance(sharp, sigma_s=0.3, sigma_r=0.01)
        clahe = cv2.createCLAHE(clipLimit=2., tileGridSize=(2, 2))
        lab = cv2.cvtColor(dst, cv2.COLOR_BGR2LAB) # convert from BGR to LAB color
space
        l, a, b = cv2.split(lab) # split on 3 different channels
        l2 = clahe.apply(l) # apply CLAHE to the L-channel
        lab = cv2.merge((l2, a, b)) # merge channels
        img2 = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
        os.chdir(str(pathroot) + "\\temp")
        cv2.imwrite("tempimg.png", img2)
        cv2.imshow('RESULT (NON-HDR) (' + str(w_int) + 'x' + str(h_int) + ')', img2)

        # FakeHDR
        prehdr = cv2.imread("tempimg.png")
        hsv = cv2.cvtColor(prehdr, cv2.COLOR_BGR2HSV)
        hsv1 = hsv.copy()

        h, s, v = cv2.split(hsv)

```

```

value1 = 75
lim = value1
v[v < lim] = 0
v[v >= lim] -= value1
mid = 0.3
mean = np.mean(v)
gamma = math.log(mid * 255) / math.log(mean)
val_gamma = np.power(v, gamma).clip(0, 255).astype(np.uint8)
low1 = cv2.merge((h, s, val_gamma))

h1, s1, v1 = cv2.split(hsv1)
value2 = 50
lim1 = 255 - value2
v1[v1 > lim1] = 255
v1[v1 <= lim1] += value2
high1 = cv2.merge((h1, s1, v1))

lowex = cv2.cvtColor(low1, cv2.COLOR_HSV2BGR)
highex = cv2.cvtColor(high1, cv2.COLOR_HSV2BGR)
cv2.imwrite("low.png", lowex)
cv2.imwrite("high.png", highex)

img_fn = ["low.png",
          "tempimg.png",
          "high.png"]

images = []
for filename in img_fn:
    im = cv2.imread(filename)
    images.append(im)
alignMTB = cv2.createAlignMTB()
alignMTB.process(images, images)
mergeMertens = cv2.createMergeMertens()
exposureFusion = mergeMertens.process(images)
cv2.imwrite("temphdr.png", exposureFusion * 255)
hdrimgpath = str(tempresfolder) + "\\temphdr.png"
hdrimg = cv2.imread(hdrimgpath)
cv2.imshow('RESULT (HDR) (' + str(w_int) + 'x' + str(h_int) + ')', hdrimg)

downloadbt["state"] = "enable"

os.chdir(pathroot)
except NameError:
    tk.messagebox.showerror(title=None, message="Please choose image (.jpg or
.png)")
return ()

def result():
    finalimg = cv2.imread(str(tempresfolder) + "\\tempimg.png")
    finalhdr = cv2.imread(str(tempresfolder) + "\\temphdr.png")
    now = datetime.now()
    dt_file = now.strftime("%d%m%Y_%H%M%S") + ".png"
    dt_file1 = "hdr_" + now.strftime("%d%m%Y_%H%M%S") + ".png"
    if not os.path.exists(str(pathroot) + "\\result"):
        os.mkdir(str(pathroot) + "\\result")
    os.chdir(str(pathroot) + "\\result")
    cv2.imwrite(dt_file, finalimg)
    cv2.imwrite(dt_file1, finalhdr)

```

```

        messagebox.showinfo("Done!", "Images saved!")
        subprocess.Popen('explorer ' + str(pathroot) + "\\result")
        return()

    icon = PhotoImage(file='icons/upload_icon.png')
    icon2 = PhotoImage(file='icons/enlarge_icon.png')
    icon3 = PhotoImage(file='icons/result_icon.png')

    upload_button = ttk.Button(
        root,
        image=icon,
        text='Upload image',
        compound=tk.LEFT,
        command=uploadimg
    )

    enlargebt = ttk.Button(
        root,
        image=icon2,
        text='Enlarge image',
        compound=tk.LEFT,
        command=enlarge
    )

    downloadbt = ttk.Button(
        root,
        image=icon3,
        text='Save image',
        compound=tk.LEFT,
        command=result
    )

    upload_button.pack(
        ipadx=20,
        ipady=5,
        fill = tk.X
    )

    enlargebt["state"] = "disabled"
    downloadbt["state"] = "disabled"

    enlargebt.pack(
        ipadx=20,
        ipady=5,
        fill = tk.X
    )

    downloadbt.pack(
        ipadx=20,
        ipady=5,
        fill = tk.X
    )

    scl.pack(
        ipadx=2,
        ipady=2,
        side=tk.LEFT,
        anchor=tk.N,

```

```
        expand=False
    )
    sc2.pack(
        padx=2,
        pady=2,
        side=tk.LEFT,
        anchor=tk.N,
        expand=False
    )
    sc3.pack(
        padx=2,
        pady=2,
        side=tk.LEFT,
        anchor=tk.N,
        expand=False
    )

    sc1["state"] = "disabled"
    sc2["state"] = "disabled"
    sc3["state"] = "disabled"
    root.mainloop()
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

ДОДАТОК Б
(обов'язковий)

КОПІЯ НАУКОВОЇ ПУБЛІКАЦІЇ

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XIV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2022»

18-19 листопада 2022

Хмельницький 2022

УДК 004:37:001:62

Збірник наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022». Хмельницький – 2022.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: apkt.khnu@gmail.com

УДК 004.932

Павлюк В.А.

*Хмельницький національний університет***МЕТОД ТА ПРОГРАМНІ ЗАСОБИ МАСШТАБУВАННЯ ЗОБРАЖЕНЬ**

Розглянуто основні алгоритми інтерполяції зображення, а також їх властивості. Один із розглянутих алгоритмів буде використаний у подальшій розробці методу масштабування зображення. Запропонований метод поєднуватиме збільшення зображення та його покращення із використанням невеликої кількості ресурсів комп'ютера.

The main algorithms of image interpolation and their properties, are considered. One of the considered algorithms will be used in the further development of the image scaling method. The proposed method will combine image enlargement and enhancement with a small amount of computer resources.

На сьогоднішній день зображення відіграють величезну роль у цифровому просторі. З кожним роком роздільна здатність екранів мобільної техніки покращується, тому зростають вимоги до роздільної здатності та якості зображень. Чим вища якість певного зображення, тим приємнішим і вагомішим стає його перегляд для користувача. Тому часто з'являється потреба у програмному масштабуванні растрового зображення. ...

Метою дослідження є розробка методу масштабування зображення, який можна буде не буде використовувати велику кількість ресурсів та буде поєднувати масштаб зображення із покращенням його кольорового тону....

Доречним буде використання інтерполяції при зміні розмірів сторін оригінального зображення. Інтерполяція дозволяє обчислити додаткові пікселі у новому зображенні. Якщо вихідне зображення є меншим, тоді збільшене зображення буде мати додаткові пікселі, які не зовсім збігаються з сусідніми пікселями [1].

Дослідження алгоритмів інтерполяції дозволить з'ясувати їх властивості, а також з'ясувати, який із методів зазнає найменше втрат у якості та підходить для подальшої розробки власного методу масштабування зображення.

У таблиці 1 описано методи інтерполяції та їх властивості.

Таблиця 1 – Методи інтерполяції та їх властивості

Алгоритм інтерполяції	Властивість
Інтерполяція методом найближчого сусіда або округла інтерполяція	Найпростіший алгоритм. Приймає округлене значення очікуваної позиції та знаходить найближче значення даних у цілочисловій позиції.
Білінійна інтерполяція	Білінійна інтерполяція є узагальненням лінійної інтерполяції для функції із двома змінними. На відміну від інтерполяції найближчого сусіда, білінійна інтерполяція виконує інтерполяцію у двох вимірах і передбачає функцію, яка використовується для обчислення кольору пікселя.
Інтерполяція співвідношенням площ	Даний алгоритм використовує повторну дискретизацію за допомогою співвідношень площ пікселів. При збільшенні не відрізняється від інтерполяції найближчого сусіда, тому цей алгоритм доцільно використовувати при зменшенні зображення
Бікубічна інтерполяція	Бікубічна інтерполяція враховує 16 пікселів (4×4 пікселі), в той час, як білінійна враховує лише 4 пікселі. Це дозволяє отримати більш згладжений результат.
Інтерполяція Ланцоша	Даний алгоритм враховує 64 сусідні пікселі (8×8) та усереднює їх за допомогою функцій sinc. Інтерполяція Ланцоша є оптимальною для зменшення дискретизації зображень, тобто його зменшення. Вона забезпечує згладжування з мінімальним розмиттям.

На рисунку 1 можна побачити графічне відображення того, як відбувається інтерполяція зображення різними алгоритмами:

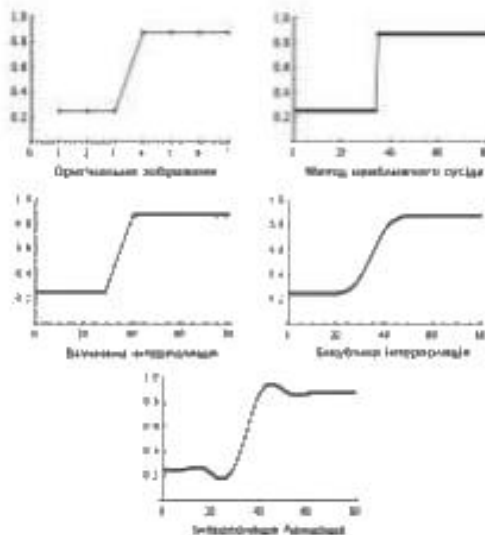


Рисунок 1 – Графічне відображення роботи алгоритмів

Наступним кроком буде порівняння результатів алгоритмів інтерполяції, що дозволить вибрати один із них для подальшої розробки власного методу масштабування зображень. На рисунку 2 зображено результати інтерполяції на прикладі текстової вивіски університету, де а – метод найближчого сусіда, b – білінійна інтерполяція, c – співвідношення площ, d – бікубічна, e – інтерполяція Ланцоша:



Рисунок 2 – Результати алгоритмів інтерполяції

Серед розглянутих алгоритмів найбільш доцільним буде використання алгоритму білінійної інтерполяції, а саме білінійну інтерполяцію з точністю до біту (LINEAR_EXACT). Саме цей алгоритм допоможе збільшити зображення із

помірним згладжуванням та низькою кількістю кольорових шумів та артефактів, які будуть у подальшому усуватись за допомогою обробки зображення.

Розроблюваний метод складатиметься з деяких кроків, для того, щоб збільшити, відновити та покращити зображення. Алгоритм роботи методу, що розробляється, показаний на рисунку 3.

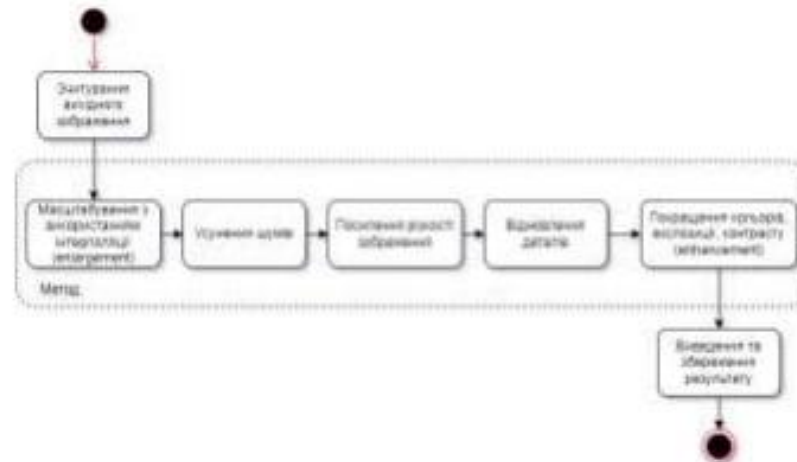


Рисунок 3 – Алгоритм роботи методу

Подальші дослідження будуть спрямовані на розробку методу масштабування зображення, який буде поєднувати у собі збільшення зображення та покращення його експозиції та кольорової гамми.

Перелік посилань

1. Image Interpolation techniques in digital image processing. URL: https://www.researchgate.net/publication/346625606_Image_Interpolation_techniques_in_digital_image_processing
2. Gonzalez RC, Woods RE. 2002. Digital image processing. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall. ISBN: 0-201-18075-8.

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Метод та програмні засоби масштабування зображень

Виконав:
студент II курсу, групи ІПЗм-21-1
Павлюк Віталій

Керівник:
доктор фіз.-мат. наук, професор
Бедратюк Леонід

Об'єкт, предмет та мета дослідження

- Об'єкт дослідження – масштабування зображень.
- Предмет дослідження – методи, алгоритми та програмні засоби масштабування растрових зображень.
- Мета дослідження – розробка методу масштабування зображення, який не буде використовувати велику кількість ресурсів та буде поєднувати збільшення роздільної здатності зображення із покращенням його кольорового тону.

Актуальність теми

- **Актуальність** теми полягає у тому, що дуже часто користувачі або розробники стикаються із растровими зображеннями низької або недостатньо високої якості, тому дуже часто вони користуються програмними продуктами, які дозволяють покращити якість та масштабувати зображення.
- Тема масштабування зображення актуальна на протязі декількох десятків років і до сих пір розробники знаходять нові рішення та методи щодо збільшення масштабу зображення із меншою кількістю втрат якості
- Більшість наявних методів використовують нейронні мережі, що призводить до зайвих навантажень на пристрій, тому виникає потреба у розробці нового алгоритму, ґрунтуючись на проблемах, які є присутніми у вже існуючих методах.

Завдання дослідження

Задля досягнення поставленої мети потрібно вирішити наступні задачі:

- проаналізувати та визначити специфіку предметної області;
- визначити основні переваги та недоліки існуючих методів масштабування зображень;
- визначити чіткі цілі для розробки методу;
- спроектувати алгоритм масштабування, обробки та покращення растрових зображень;
- здійснити програмну реалізацію методу;
- провести тестування програмного продукту.
- проаналізувати та дослідити отримані результати.

Аналіз наявних проблем та можливих їх рішень

Одними з основних проблем у галузі є:

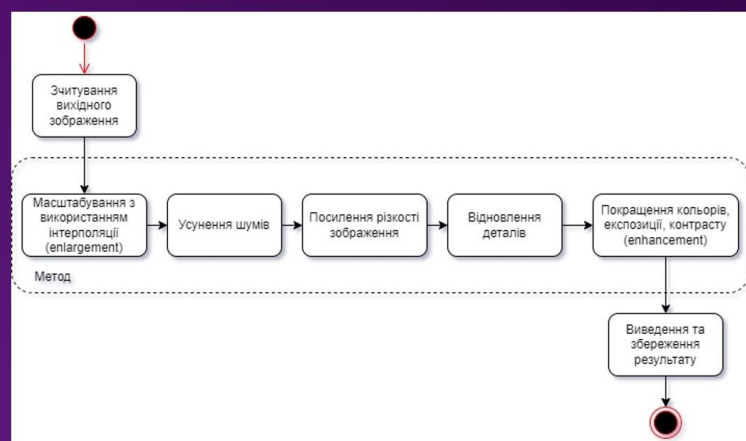
- використання надмірної кількості ресурсів;
- високі вимоги до апаратних компонентів;
- втрата деталей (таких як текст і т.д.).

Таким чином з'являється потреба у створенні методу масштабування зображення, який задовольнятиме наступні критерії:

- швидкодія;
- низька ресурсовитратність;
- помірні вимоги до апаратних пристроїв;
- простота у використанні.

Розроблені методи

- Для розробки методу та програмного засобу для масштабування цифрового зображення було розроблено алгоритм, який показує структуру та алгоритм роботи методу та програмного засобу масштабування зображень.



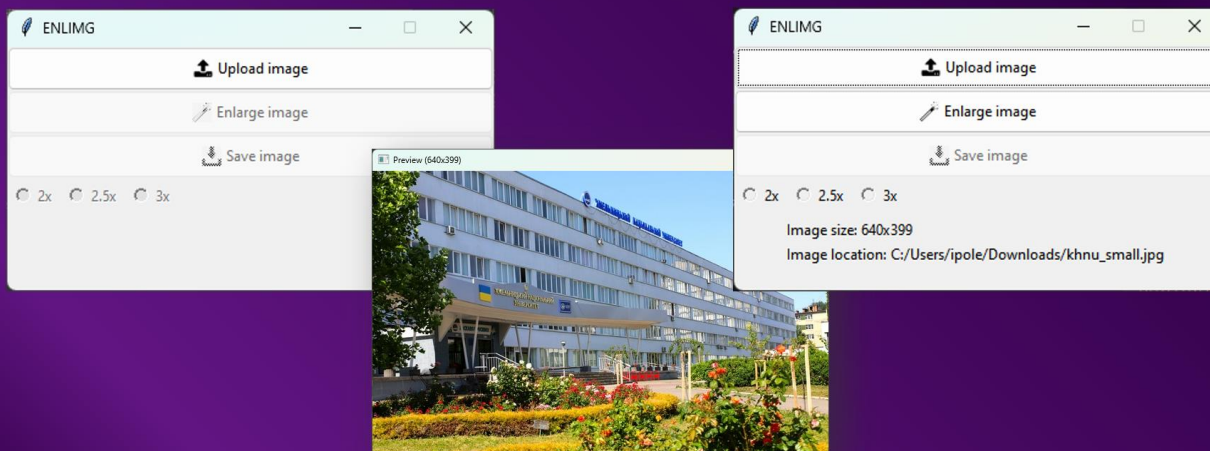
Розроблені методи

- Додатково було розроблено метод **Fake-HDR**, завданням якого є коригування динамічного діапазону зображення. Такий метод **HDR** є не дуже повноцінним через часту появу артефактів при використанні **JPEG**, в той час, як справжній **HDR** використовує зйомку **RAW** із камери.



Практична імплементація

- Розроблений програмний продукт має простий та інтуїтивно зрозумілий для користувача інтерфейс, який буде економити ресурси пристрою користувача, що позитивно впливатиме на продуктивність методу та персонального комп'ютера користувача в цілому.
- Також реалізовано збереження результатів із **Fake-HDR** алгоритмом та без нього, що позитивно вплине на користувача та його досвід користування програмним продуктом.



Апробація результатів

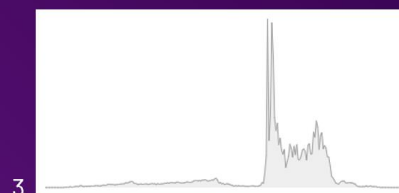
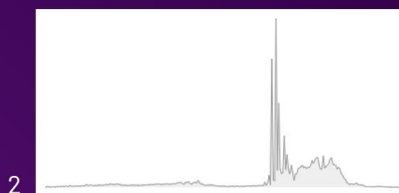
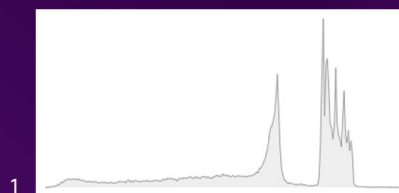
- Для підтвердження ефективності та працездатності розробленого методу та програмного засобу використано метрики порівняння **SSIM, MS-SSIM, VIF** та **UQI**
- Результати показують, що метод повністю задовольняє умови схожості із оригінальним зображенням

```

MS-SSIM score (ORIGINAL - NON-HDR): (0.909119213901273+0j)
SSIM score (ORIGINAL - NON-HDR): (0.8073058863496456, 0.8268789134550741)
VIF Index (ORIGINAL - NON-HDR): 0.32856901290918433
UQI score (ORIGINAL - NON-HDR): 0.9318878986423188
MS-SSIM score (ORIGINAL - HDR): (0.9167504595547992+0j)
SSIM score (ORIGINAL - HDR): (0.7947657483389324, 0.8252543813304274)
VIF score (ORIGINAL - HDR): 0.31467733647072893
UQI score (ORIGINAL - HDR): 0.907468344967143
MS-SSIM score (NON-HDR - HDR): (0.9904896563356189+0j)
SSIM score (NON-HDR - HDR): (0.9727385412827158, 0.9848712333550572)
VIF Index (NON-HDR - HDR): 0.9999999999877606
UQI score (NON-HDR - HDR): 0.9693191841644326
  
```

Апробація результатів

- При порівнянні результатів було з'ясовано, що результат алгоритм Fake-HDR (3) має більшу кількість деталей, ніж звичайний (2) метод.
- Також гістограма освітленості показує, що зображення з HDR має більшу кількість середніх тонів



Наукова новизна

- вперше запропоновано метод масштабування зображення, що відрізняється від наявних і дозволяє уникнути використання нейронних мереж, що допоможе збільшити швидкість обробки зображення при набагато меншій кількості ресурсів;
- вперше запропоновано метод коригування динамічного діапазону зображення з використанням експозиційного синтезу та злиття зображень, що забезпечує рівномірність між темними темними та світлими ділянками зображення.

Практичне значення

- Практична цінність отриманих результатів полягає у успішній розробці методу та програмного засобу масштабування зображень. Завдяки поліпшеним характеристикам, у порівнянні з традиційними рішеннями, розроблена програмна система має високі конкурентні шанси на ринку.

Наукові публікації

Опубліковано тези у збірнику тез XIV Всеукраїнської науково-практичної конференції:

Павлюк В. А. Метод та програмні засоби масштабування зображення //XIV Всеукраїнська науково-практична конференція "Актуальні проблеми комп'ютерних наук (АПКН - 2022). - м. Хмельницький: 2022

Висновки та рекомендації

У результаті виконання дипломної роботи було вирішено наступні задачі:

- проаналізовано та визначено специфіку предметної області;
- визначено основні переваги та недоліки існуючих методів масштабування зображень;
- визначено чіткі цілі для розробки методу;
- спроектовано алгоритм масштабування, обробки та покращення растрових зображень;
- здійснено програмну реалізацію методу;
- проведено тестування програмного продукту.
- проаналізовано та досліджено отримані результати.

Результати практичної апробації програмної системи підтверджують її працездатність. Тому її можна використовувати у компаніях, які здійснюють обробку зображень чи мають потребу у цьому.

Дякую за увагу!



Завідувачу кафедри інженерії програмного забезпечення проф. Бедратюк Л. П.
здобувача вищої освіти
Павлюка В. А.
факультет ІІ, 2 курс, група ПІЗм-21-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

29.11.2022 р.

підпис

підпис

Thu Dec 01 08:57:59 EET 2022, Хирів Володимир Русланович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 14.0%

Словники перевірки: en_US, ru_RU, ua_UA Помилки в документах: 11%

ID: 108855 Назва: КРМ на тему: Метод та програмні засоби масштабування зображення Додано в БД: 2022-12-01 Автора: Павлюк В.А. Керівника: Бедратюк Л.П. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	70266	603	14066 (20%)	150 (25%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
107859	Назва: Метод та програмні засоби масштабування зображення. Додано в БД: 2022-10-04 Автора: В.А. Павлюк Керівника: Бедратюк Л.П. Консультанти: Опоненти:	9911 (14.0%)	101 (17.0%)



Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1013117533

Дата перевірки:
01.12.2022 09:52:04 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
02.12.2022 09:26:32 EET

ID користувача:
100005589

Назва документа: diploma_master_full_без дод

Кількість сторінок: 78 Кількість слів: 10402 Кількість символів: 86407 Розмір файлу: 11.12 MB ID файлу: 1012873705

10.9% Схожість

Найбільша схожість: 6.87% з джерелом з Бібліотеки (ID файлу: 1009383373)

3.66% Джерела з Інтернету

213

Сторінка 80

8.51% Джерела з Бібліотеки

160

Сторінка 81

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Метод та програмні засоби масштабування зображень»

Автор: Павлюк Віталій Андрійович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Леонід Петрович, доктор фіз.-мат. наук, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті дипломної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, назвах розділів/підрозділів тощо та в назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає **10,9%** і адресується до 213 джерел з Інтернету та 160 джерел із бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломної роботи.

Керівник



Л. П. Бедратюк

Гарант ОП



О. М. Яшина

Завідувач кафедри



Л. П. Бедратюк

СУПРОВІДНИЙ ЛИСТ

Назва роботи	Метод та програмні засоби масштабування зображень
Автор	Павлюк Віталій Андрійович
Тематика	121 Інженерія програмного забезпечення
Ключові слова	масштабування зображень
	обробка зображень
	широкий динамічний діапазон
	ядра зображення
Дата публікації	2022
Видавництво	Хмельницький національний університет
Бібліографічний опис	Павлюк В. А. Метод та програмні засоби масштабування зображень: дипломна робота магістра : 121 Інженерія програмного забезпечення / В. А. Павлюк ; Хмельниц. нац. ун-т. – Хмельницький, 2022. – 95 с.
Короткий огляд (реферат)	<p>Мета дослідження – розробка методу масштабування зображення, який не буде використовувати велику кількість ресурсів та буде поєднувати збільшення роздільної здатності зображення із покращенням його кольорового тону. Для програмної реалізації було використано мову програмування Python, модулі OpenCV та Pillow, а також Tkinter для розробки графічного інтерфейсу. Для проведення тестування було використано метрики схожості зображень. Проведені емпіричні випробування розробленого методу довели його придатність і ефективність у порівнянні з конкуруючими рішеннями у галузі масштабування та обробки зображень, а також життєздатність і функціональність програмного забезпечення, створеного з його використанням. Апробація отриманих результатів показала, що результати роботи методу не зазнають викривлень та спотворень під час масштабування та обробки, а також показала, що на виконання методу програмне забезпечення витрачає від 1 до 2,2 секунд, в залежності від коефіцієнту масштабування, при цьому середня різниця у часі при збільшенні коефіцієнту масштабування дорівнює приблизно 0,25-0,3 секунди для кожного з методів або 0,5-0,7 секунд у загальному. Розроблений метод можна рекомендувати до використання компаніям, які займаються обробкою зображень та графічним дизайнерам.</p>
Тип вмісту	Магістерські роботи
Кафедра	Інженерії програмного забезпечення

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

освітнього ступеня «магістр»

Магістр Павлюк Віталій АндрійовичТема Метод та програмні засоби масштабування зображеньСпеціальність 121 «Інженерія програмного забезпечення»

Обсяг дипломної роботи:

Кількість сторінок дипломної роботи 95.

1. Короткий зміст роботи та прийнятих рішень У дипломній роботі проведено аналіз предметної області масштабування зображення. Покращено моделі та механізми створення методу та програмного засобу масштабування зображень, які б допомогли покращити та оптимізувати процес масштабування та покращення зображень. На основі розроблених моделей та механізмів імплементовано метод та програмний засіб масштабування зображення, проведено практичну апробацію та доведено його ефективність.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота освітнього ступеня «магістр» у повній мірі відповідає поставленому завданню як у теоретичній, так і в практичній її частині

2. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи У вступі обґрунтовується актуальність теми роботи, формулюються цілі і завдання дослідження, описується наукова новизна та практична значимість отриманих результатів. У першому розділі охарактеризовано структуру предметної області та існуючі функціональні моделі і методи організації методу та програмного засобу масштабування зображення, виконана розгорнута постановка задачі. У другому розділі досліджено методи і способи вирішення поставлених задач. Традиційні моделі системи, досліджені на етапі аналізу, були розширені новими методами та засобами з метою покращення базових характеристик системи. У третьому розділі обґрунтовано проектні рішення, що дають змогу реалізувати технічні вимоги, забезпечити сумісність та взаємодію різних компонентів системи. У четвертому розділі розглянуто питання, що стосуються реалізації програмної системи на основі прийнятих проектних рішень, а також її технічні та технологічні характеристики. Також проведено емпіричне дослідження, спрямоване на доведення працездатності розробленої системи та її функціональної придатності. Обґрунтована ефективність створених методів та засобів та розроблено рекомендації з їх застосування при проектуванні методу та програмного засобу масштабування зображення.

4. Позитивні сторони роботи Дипломна робота містить низку інноваційних рішень, зокрема, розроблений метод для покращення динамічного діапазону зображення, розроблено метод масштабування зображення з використанням інтерполяції та оптимізовано його роботу.

5. Негативні сторони роботи Розроблений метод покращення широкого динамічного діапазону зображення може призводити до появи невеликих артефактів, які з'являються через різницю рівнів яскравості зображень при використанні зображення з надмірно низькою роздільною здатністю.

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми дипломної роботи з дотриманням вимог стандартів. Пояснювальна записка відповідає вимогам стандартів до її оформлення.

7. Відгук про роботу в цілому В цілому дипломна робота заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи є послідовними та логічними, що дозволяє чітко розуміти викладений матеріал у рамках тематики дипломної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для вирішення поставленої задачі.

8. Інші зауваження _____

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що вона заслуговує оцінки «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Товарущенко Тетяна Александрівна, доктор технічних наук, професор, зав. кафедри комп'ютерної інженерії та інформаційних систем (КПІС) ХНУ

« 26 » 11 2022 р.


(підпис)