Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:

«РЕАЛІЗАЦІЯ КОМПЛЕКСНОЇ СИСТЕМИ ОБРОБКИ ЗАШУМЛЕНИХ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ АДАПТИВНИХ ФІЛЬТРІВ НА PYTHON»

Виконав: студент IV курсу, групи 1ІСТ-21б
спеціальності 126 – Інформаційні системи та технології
(шифр і назва спеціальності)
Бодян І.С.
(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри АІІТ
Софина О.Ю.
(прізвище та ініціали)
«11» 06 2025 р.

Рецензент: к.т.н., доцент кафедри САІТ
Крижановський Є. М.
(прізвище та ініціали)
«11» 06 2025 р.

Допущено до захисту
завідувач кафедри АІІТ
«12» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 126 – Інформаційні системи та технології
Освітньо-професійна програма Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ
завідувач кафедри АІІТ
_____ д.т.н., проф. Бісікало О. В.
«__» 06 2025 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Бодян Ірині Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи «Реалізація комплексної системи обробки зашумлених зображень з використанням адаптивних фільтрів на Python»
керівник роботи Софина Ольга Юріївна, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «20» березня 2025 року №97
2. Термін подання студентом роботи 10.06.2025 р.
3. Вихідні дані до роботи:
- використання графічного інтерфейсу (GUI) для взаємодії з користувачем;
- підтримка обробки зображень із трьома типами шуму;
- можливість застосування трьох типів фільтрів;
- інтеграція модулів для візуального порівняння результатів до та після фільтрації;

4. Зміст текстової частини (перелік питань, які потрібно розробити) провести огляд видів шумів та їх впливу на якість цифрових зображень, проаналізувати існуючі методи та обґрунтувати вибір адаптивних фільтрів, розробити та реалізувати архітектуру програмної системи фільтрації зашумлених зображень, модулі генерації шуму, адаптивної фільтрації та графічного інтерфейсу
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) Блок-схема алгоритму роботи програми; тестування різними методами зашумлення зображень; тестування різними методами фільтрації зображень; схема інтеграції з зовнішніми системами

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Софина О.Ю., доц. каф. АІІТ		

7. Дата видачі завдання 20.03.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Термін виконання		Примітки
		початок	закінчення	
1	Вибір, узгодження та затвердження теми БКР	03.10.2024	09.10.2024	вик.
2	Аналіз предметної області та опрацювання літературних джерел.	12.03.2025	29.03.2025	вик.
3	Постановка задач для вирішення в БКР	01.04.2025	26.04.2025	вик.
4	Вирішення поставлених задач	29.04.2025	10.05.2025	вик.
5	Підтвердження достовірності отриманих результатів	13.05.2025	24.05.2025	вик.
7	Оформлення пояснювальної записки та графічної частини	27.05.2025	31.05.2025	вик.
8	Перевірка ПЗ на відповідність вимогам	03.06.2025	07.06.2025	вик.
9	Попередній захист, доопрацювання, та рецензування БКР	10.06.2025	15.06.2025	вик.
10	Захист БКР ЕК	18.06.2025	18.06.2025	вик.

Студент _____ Бодян І. С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Софина О. Ю.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 88 сторінок формату А4 в тому числі додатки, на яких є 22 рисунків, 6 таблиць, список використаних джерел, що містить 25 найменувань.

У роботі розглядається процес розробки програмної системи для обробки зашумлених зображень з використанням адаптивних фільтрів мовою програмування Python. Метою дослідження є підвищення якості зображень шляхом ефективного видалення шумів із збереженням важливих деталей. У роботі проаналізовано різни типи шумів, зокрема гаусівський, імпульсний та рівномірний шум, а також обґрунтовано вибір відповідних адаптивних фільтрів, таких як медіанний, вейвлет-фільтр та адаптивний фільтр Вінера. Запропоновано архітектуру системи, реалізовано модулі попередньої обробки, фільтрації та візуалізації результатів.

Ключові слова: зашумлене зображення, адаптивні фільтри, фільтр Вінера, медіанний фільтр, вейвлет фільтр, імпульсний шум, гаусівський шум, рівномірний шум, обробка зображення, комп’ютерний зір.

ANOTATION

The bachelor's qualification work consists of 88 pages of A4 format, including appendices, which contain 22 figures, 6 tables, a list of used sources, containing 25 names.

The work considers the process of developing a software system for processing noisy images using adaptive filters in the Python programming language. The purpose of the research is to improve image quality by effectively removing noise while preserving important details. The work analyzes various types of noise, in particular Gaussian, impulse and uniform noise, and also justifies the choice of appropriate adaptive filters, such as median, wavelet filter and adaptive Wiener filter. The system architecture is proposed, modules for preprocessing, filtering and visualization of results are implemented.

Keywords: noisy image, adaptive filters, Wiener filter, median filter, wavelet filter, impulse noise, Gaussian noise, uniform noise, image processing, computer vision.

ЗМІСТ

ВСТУП..... 4
1 ТЕОРИТИЧНІ ОСНОВИ ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА ФІЛЬТРАЦІЇ ШУМІВ..... 6
1.1 Поняття цифрового зображення. 6
1.2 Види шумів у зображеннях: гаусівський, імпульсний, рівномірний. ... 8
1.3 Методи виявлення шумів у зображеннях. 13
1.4 Класичні методи фільтрації шуму: середній, медіанний, гаусівський фільтр. 14
1.5 Адаптивні фільтри: принцип дії, класифікація та особливості. 16
2 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ……21
2.1 Визначення функціональних вимог до системи обробки зображень. . 21
2.2 Вибір типів шумів для дослідження та фільтрації. 22
2.3 Обґрунтування вибору адаптивних фільтрів. 25
2.4 Вибір мови програмування та бібліотек. 30
2.5 Архітектура програмної системи: модулі, взаємодія компонентів. 35
2.6 Формалізація критеріїв якості фільтрації (PSNR, SSIM)..... 38
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ТА ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ 41
3.1 Реалізація інтерфейсу завантаження та попереднього перегляду зображень. 41
3.2 Модуль додавання штучного шуму для тестування фільтрів. 43
3.3 Реалізація адаптивних фільтрів. 46
3.4 Візуалізація результатів фільтрації: «до» та «після». 48
3.5 Оцінювання якості відновлення зображень з використанням метрик. 53
4ТЕСТУВАННЯ, ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ 57

4.1 Порівняння ефективності фільтрів за обраними метриками. 57
4.2 Аналіз переваг та недоліків використаних фільтрів. 60
4.3 Аналіз придатності системи до різних типів зображення. 63
4.4 Пропозиції щодо розширення системи. 66
4.5 Перспективи використання в практичних задачах. 69
ВИСНОВКИ 72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 74
ДОДАТКИ 78
Додаток А (обов'язвоковий) ІЛЮСТРАТИВНА ЧАСТИНА 79
Додаток Б (обов'язковий) Лістинг основних функцій програми 83
Додаток В (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ ..... **Помилка! Закладку не визначено.**

ВСТУП

У сучасну цифрову епоху зображення є основним джерелом візуальної інформації у медицині, промисловості, безпеці, дистанційному зондуванні та комп'ютерному зорі. Проте при отриманні, передачі чи обробці зображень часто виникають спотворення від шуму, що ускладнює інтерпретацію, знижує ефективність аналізу та робить подальшу обробку неточною. Це зумовлює необхідність створення інструментів для якісного очищення зображень без втрати важливих деталей.

Найбільш складними є три види шуму — гаусівський, імпульсний і рівномірний. Вони мають різний вплив на зображення, тому вимагають окремих підходів до фільтрації. Класичні універсальні фільтри часто не дають належного результату, особливо в умовах складного або змішаного шуму. Тому актуальним є створення адаптивної системи, яка зможе розпізнавати тип шуму й автоматично підбирати відповідний метод його усунення.

У рамках цієї бакалаврської кваліфікаційної роботи було розроблено та реалізовано комплексну систему фільтрації зашумлених зображень за допомогою мови програмування Python. Вона дозволяє ефективно видаляти різні типи шуму зі збереженням критично важливої візуальної інформації. Основні можливості системи включають:

- Використання трьох адаптивних фільтрів: медіанного, Вінера та вейвлет-фільтра;
- Інтеграцію модулів для генерації шуму та візуального порівняння обробки;
- Наявність графічного інтерфейсу для зручного завантаження, обробки та аналізу зображень;
- Автоматичну оцінку ефективності очищення за допомогою метрик PSNR та SSIM.

Отримані результати підтверджують високу ефективність впроваджених фільтрів для обробки відповідних типів шуму. Зокрема, використання

медіанного фільтра сприяло покращенню якості обробки імпульсного шуму на 87% за показником SSIM порівняно з вихідним зашумленим зображенням. Фільтр Вінера зарекомендував себе як ефективний засіб відновлення зображень із гаусівським шумом, зберігаючи основну структуру без суттєвих змін. Вейвлет-фільтр, своєю чергою, продемонстрував найвищу ефективність у роботі з рівномірним та змішаним шумом.

Основним науково-практичним результатом роботи є розробка комплексної адаптивної системи фільтрації зашумлених зображень, яка здатна забезпечити високий рівень якості відновлення візуальної інформації при мінімальному втручанні користувача.

Метою роботи являється підвищення якості цифрових зображень шляхом ефективного видалення різноманітних видів шумів, зберігаючи при цьому важливі деталі візуальної інформації

Для досягнення поставленої мети необхідно **виконати наступні завдання**:

1. Проаналізувати типи шумів і методи їх фільтрації.
2. Обґрунтувати вибір адаптивних фільтрів для різних типів шумів.
3. Реалізувати алгоритми фільтрації в середовищі Python.
4. Провести тестування ефективності реалізованої системи.

Об’єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи адаптивної фільтрації зашумлених зображень.

Методи дослідження – використання математичної статистики, цифрової обробки зображень, а також програмні засоби Python та бібліотеки OpenCV, NumPy, scikit-image для реалізації фільтрів і проведення експериментів.

Апробація результатів дослідження була здійснена шляхом представлення доповіді на LIV Всеукраїнській науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (2025) [9].

1 ТЕОРИТИЧНІ ОСНОВИ ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА ФІЛЬТРАЦІЇ ШУМІВ

1.1 Поняття цифрового зображення.

Цифрове зображення являє собою матрицю числових значень, яка описує інтенсивність світла в кожній точці (пікселі) простору. Ці числові дані утворюються шляхом дискретизації аналогового сигналу, отриманого з оптичного пристрою, наприклад, камери або сканера. Кожен піксель має свої координати та значення яскравості чи кольору, що залежить від формату зображення.

Залежно від обсягу інформації, що зберігається в кожному пікселі, цифрові зображення поділяються на різні типи:

- Бінарні зображення: містять лише два типи значення пікселів – 0 (чорний) і 1 (білий).
- Зображення у градаціях сірого: кожен піксель відображає інтенсивність світла, зазвичай у діапазоні від 0 до 255.
- Кольорові зображення: складаються із трьох каналів — червоного (Red), зеленого (Green) і синього (Blue), кожен із яких теж має значення у межах від 0 до 255. Завдяки комбінації цих значень можливо створити мільйони відтінків.
- Мультиспектральні або гіперспектральні зображення: включають більше трьох спектральних каналів і використовуються для спеціалізованих задач, як-от супутникова зйомка чи медична візуалізація.

Технічно цифрове зображення представлено у вигляді масиву, що може бути дво- або тривимірним:

- Розмірність: визначає ширину й висоту зображення, вимірювані в кількості пікселів.

- Глибина кольору: визначає кількість бітів для кодування одного пікселя. Наприклад, 8-бітне зображення дозволяє передавати 256 рівнів сірого, а 24-бітне кольорове — понад 16 мільйонів комбінацій кольорів.

Формально цифрові зображення можна зобразити як матриці:

- Сіре зображення: це двовимірна матриця розміром M × N, де M — висота, N — ширина.

- Кольорове зображення: представлено тривимірною матрицею M × N × 3, де три канали відповідають компонентам RGB (табл.1.1).

Така структура дозволяє ефективно кодувати, зберігати й обробляти візуальну інформацію.

Таблиця 1.1 – Формати зображень.

Формат	Характеристика	Підтримка стиснення
BMP	Не стиснутий, великий розмір.	Ні
JPEG	Стиснення з втратою якості, популярний для фото	Так.
PNG	Стиснення без втрат, підтримка прозорості.	Так.
TIFF	Висока якість, підтримка шарів і каналів.	Може бути з втратами або без
GIF	Обмежена палітра (256 кольорів), підтримка анімації.	Так
WEBR	Ефективне стиснення, створене Google.	Так

Вибір формату зображення залежить від поставленої мети: зберігання із максимальною якістю, оптимізація для передачі через мережу, підтримка прозорості та інших вимог.

У програмуванні зазвичай представляють як багатовимірні масиви, наприклад, у Python для цього використовується бібліотека NumPy. Наприклад:

```
import cv2
img = cv2.imread("image.jpg")
```

```
print(img.shape)  # (height, width, 3)
```

Такий підхід забезпечує зручний доступ до кожного пікселя для подальшої обробки за допомогою математичних методів. Це створює основу для реалізації фільтрації, аналізу та інших завдань у роботі з зображеннями.

1.2 Види шумів у зображеннях: гаусівський, імпульсний, рівномірний.

Шум у зображеннях – це випадкові або псевдовипадкові, відхилення яскравості чи кольору пікселів, які не мають відношення до зображуваного об’єкта. Він помітно знижує якість зображення, ускладнює аналіз і може викривляти результати обробки, розпізнавання чи класифікації. Не дивлячись на те, що існує безліч різновидів шуму, три з них – гаусівський, імпульсивний та рівномірний – є основоположниками для розуміння цього явища.

Гаусівський шум, також відомий як нормальний або білий шум, є одним із найпоширеніших типів випадкових спотворень у цифрових зображеннях. Його принцип базується на функції Гауса, що описує нормальний розподіл ймовірностей (рис. 1.1). Це значить, що більшість значень шуму знаходиться близько до нуля, а значні відхилення трапляються зрідка.

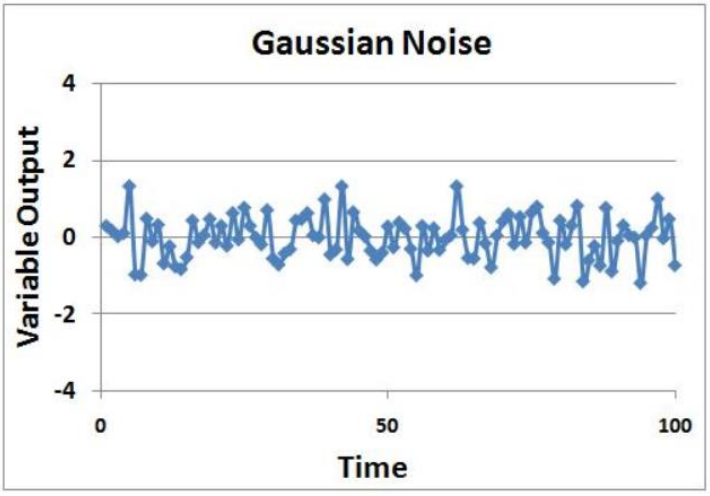

Рисунок 1.1 – Амплітуда частоти гаусівського шуму.

Такий шум зазвичай виникає через середньо-квадратичне відхилення електронних сигналів у сенсорах камер, особливо при недостатньому освітленні. Він впливає на кожен піксель незалежно, додаючи до його яскравості випадкове значення. Візуально гаусівський шум проявляється як тонке зерно або слабке мерехтіння на всій площині зображення. Невелика інтенсивність шуму найчастіше непомітна людському оку, але зі збільшенням його рівня падає чіткість і контраст, а дрібні деталі розмиваються.

Головною проблемою гаусівського шуму є його нерегулярність, що ускладнює його фільтрацію. Видалення такого шуму без втрат важливої інформації потребує тонкого підходу. Найчастіше використовуються фільтри середнього або гаусівського згладжування, які часто приглушують шум, але знижують різкість країв і текстур.

Даний шум стає серйозною проблемою в сферах медичної візуалізації, супутникового зондування чи робототехніки, де кожен піксель має критичне значення. Висока щільність цього шуму може призвести до недостовірних діагнозів або неправильних дій автономних систем. Тому моделювання гаусівського шуму і створення ефективних алгоритмів для його усунення залишаються важливими для досліджень.

Інший поширений тип шуму – імпульсивний шум, часто званий «сіллю і перцем». Його назва говорить сама за себе: такі шуми виглядають як випадкові поодинокі точки яскраво-білого або чорного кольору, які різко контрастують із навколишніми ділянками.

Імпульсний шум зазвичай виникає через помилки під час передачі даних, апаратні несправності чи дефекти сенсорної матриці. На відміну від гаусівського шуму, він не охоплює загальний фон зображення, а проявляється окремими точками.

Імпульсний шум, не зважаючи на свою локальність, є особливо помітним і дратівливим для сприйняття. Навіть кілька яскравих артефактів здатні зіпсувати загальне враження від зображення. Основна проблема полягає у високому амплітудному контрасті цього виду зашумлення: він проявляється

вкрай різким значенням – або повністю чорним, або повністю білим. По цій причині традиційні фільтри, такі як середньозважені або гаусівський, зазвичай неефективні щодо його усунення.

Функція щільності розподілу ймовірностей імпульсного шуму задається виразом

$$p(z) = P_a \text{ при } z = a;\ P_b \text{ при } z = b; 0 \text{ інших випадках.} \qquad (1.1)$$

Якщо $b > a$, то піксель з яскравістю b має вигляд світлової точки на зображенні, натомість піксель з яскравістю a виглядає навпаки, як темна точка. Якщо одне зі значень ймовірностей P_a або P_b дорівнює 0, то імпульсний шум називається уніполярним.

Для боротьби з імпульсним шумом найкраще себе зарекомендували себе медіанні фільтри, які замінюють значення пікселя на медіану його оточення. Завдяки нечутливості до екстремальних величин, такі фільтри ефективно видаляють артефакти імпульсного шуму, водночас зберігаючи межі та структуру зображення. Однак при підвищенні рівня шуму навіть медіанні фільтри потребують удосконалення, зокрема через використання адаптивних методів із змінними вікнами або алгоритмами класифікації.

Особливо складною проблемою імпульсного шуму в документах, штрих-кодах та QR-кодах, де будь яке спотворення може призвести до втрати критичної інформації. Він також негативно впливає на біометричні системи, які залежать від точності розпізнавання деталей контуру чи меж.

Третій важливий тип – рівномірний шум, тобто тип шуму, значення якого коливаються випадковим чином у задньому діапазоні з однаковою ймовірністю. Це означає, що зміни яскравості з однаковою ймовірністю відбуватимуться як у позитивному, так і в негативному напрямках, і жодному одному значенню не надається перевага.

Рівномірний шум зазвичай спричинений квантовими ефектами, недосконалістю аналого-цифрового перетворення або зовнішнім

електромагнітним впливом. Він може бути менш помітним, ніж гаусівський або імпульсивний, але в той же час він має здатність «розмивати» дрібні деталі зображення та руйнувати пропорції кольорів і текстур.

Цей тип шуму характеризується рівномірним розподілом випадкових величин, тобто він не має піку в центрі як гаусівський шум. Тому фільтрація ускладняється – ви не можете просто «заглушити» найбільш ймовірне значення. Для зменшення рівномірного шуму зазвичай використовуються адаптивні фільтри, вейвлет-аналіз або методи машинного навчання, які враховують навколишнє середовище пікселя.

Функція щільності розподілу ймовірностей рівномірного шуму задається виразом

$$p(z) = \frac{1}{b-a} \text{ при } a \leq z \leq b; \text{ 0 у всіх інших випадках} \quad (1.2)$$

Середнє значення для цього розподілу дорівнює

$$\mu = \frac{a+b}{2}, \quad (1.3)$$

а дисперсія

$$\sigma^2 = \frac{(b-a)^2}{12} \quad (1.4)$$

У реальному світі рівномірний шум рідко зустрічається в чистому вигляді, але він поширений як компонент складного шуму. Наприклад, цифрова камера з неякісним датчиком створює зображення з сумішшю гаусівського і однорідного шумів, що ускладнює подальшу обробку.

У реальному світі рівномірний шум рідко зустрічається в чистому вигляді, але він поширений як компонент складного шуму. Наприклад, цифрова

камера з неякісним датчиком створює зображення з сумішшю гаусівського і однорідного шумів, що ускладнює подальшу обробку (рис. 1.2).

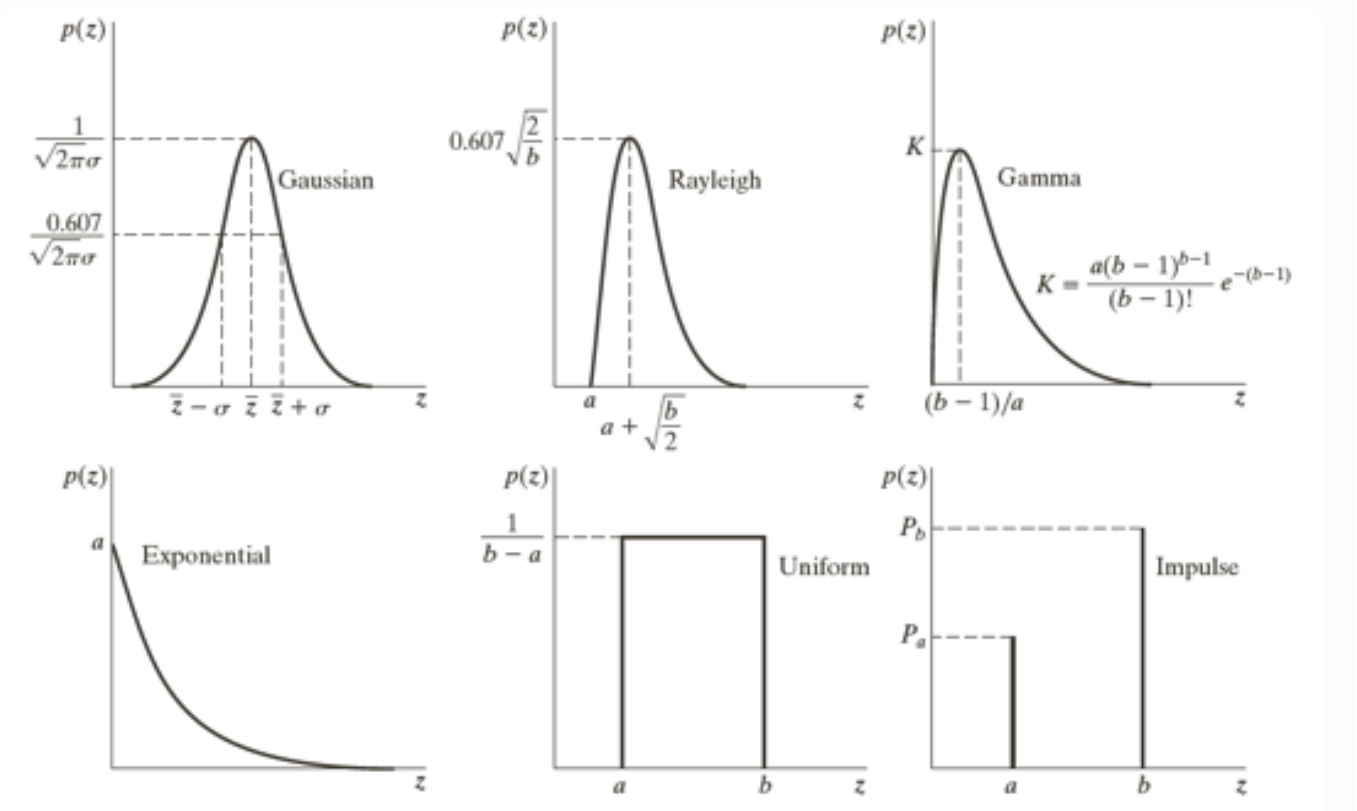

Рисунок 1.2 – Деякі важливі функції щільності розподілу ймовірностей.

Кожен з розглянутих шумів – гаусівський, імпульсний та рівномірний, має свою природу. Механізм зовнішнього вигляду та візуальних ефектів. Їх вплив на зображення змінюється від нижньої обробки до позитивних творінь, і кожен потребує особливого підходу для виявлення, аналізу та фільтра. Глибоке розуміння характеристик шуму дозволяє ефективно довести алгоритми обробки зображень, зберігати важливу інформацію та забезпечити необхідну якість для сучасних цифрових систем – від фотографії до медичних сканерів, від супутникової навігації до автономного водіння. Хоча цифровий світ не може існувати без шуму, його контрольне мистецтво полягає в тому, щоб чітко бачити мистецтво через хаос.

1.3 Методи виявлення шумів у зображеннях.

Процес виявлення шуму є важливим підготовчим етапом, що передує застосуванню методів фільтрації. Основною його метою є визначення областей зображення, які зазнали спотворень, що дозволяє адаптивно використовувати фільтри виключно на зашумлених ділянках. Ефективність подальшого усунення шуму напряму залежить від точності його детекції.

Існує кілька підходів визначення шуму, що базуються на аналізі статистичних та спектральних характеристик зображення:

1. Локальні статистичні характеристики (середнє значення та дисперсія). Найпростішим методом є оцінювання локальних змін інтенсивності пікселів. Для однорідних областей зображення (наприклад, фон, небо чи стіни) характерною є схожість сусідніх пікселів. Натомість значна варіативність цих значень може свідчити про присутність шуму в конкретному регіоні.

2. Аналіз гістограми. Гістограма є графічним методом відображення розподілу яскравостей пікселів на зображенні. Для зображень, які містять імпульсний шум, характерним є наявність виражених піків на крайніх значеннях (0 і 255). Така особливість дозволяє легко ідентифікувати присутність цього типу шуму. Гаусівський шум, навпаки, характеризується розширеною та розмитою формою гістограми, що дає змогу диференціювати його серед інших типів шуму.

3. Аналіз частотної області (перетворення Фур’є). Перехід у частотну область за допомогою перетворення Фур'є дозволяє визначати високочастотні компоненти, які властиві різноманітним видам шуму. Зокрема:

- Спектральна щільність потужності демонструє енергетичний вміст залежно від частоти сигналу.
- Наявність різкого зростання енергії на високих частотах вказає на присутність шуму в зображенні.

4. Детектори шуму на основі градієнтів. Методи визначення країв, такі як оператори Собеля, Прюітта чи Робертса, можуть бути ефективними й

для оцінки рівня шуму. Виявлення високих градієнтів у зонах, які зазвичай не містять країв, свідчить про присутність шуму.

5. Методи, що базується на ентропії. Ентропія відображає рівень випадковості у певній локальній області зображення. Зашумлені ділянки характерно вирізняються підвищеним значенням ентропії.

6. Машинне навчання та нейромережі. Основу сучасних підходів до обробки зображень складають нейромережі та машинне навчання, зокрема в задачах виявлення шуму. Сучасні алгоритми здатні навчатися розрізняти зашумлені ділянки в наборі зображень, використовуючи попередньо розмічені дані. Наприклад, нейронна мережа може визначати, які пікселі або області картинки є чистими, а які – зашумленими.

Оптимальний вибір методу залежить від типу шумового впливу, специфіки структури оброблюваного зображення та цілей подальшої фільтрації даних. У рамках досліджень використовуються як класичні статистичні методи, так і візуальний аналіз впливу шуму та ефективності його усунення. В автоматизованих рішеннях найефективніше комбінувати декілька підходів, що дозволяє досягати максимальної точності й продуктивності.

1.4 Класичні методи фільтрації шуму: середній, медіанний, гаусівський фільтр.

Фільтрація шуму є важливим етапом цифрової обробки зображень, особливо при аналізі даних медичного, супутникового або технічного характеру. Класичні методи фільтрації базуються на використанні базових математичних операцій у межах локального вікна пікселів, що дозволяє виконувати згладжування або усунення шуму. Найбільш поширеними серед них є середній (лінійний) фільтр, медіанний фільтр та гаусівський фільтр.

Середній або овзний фільтр — це лінійний метод, який виконує заміну значення окремого пікселя середнім арифметичним значенням пікселів у заданому локальному вікні.

$$I`(x,y) = \frac{1}{N}\sum_{(i,j)\in\Omega} I(x+i, y+j) \quad (1.5)$$

де Ω – локальне вікно, N – кількість пікселів у вікні.

Переваги: простота реалізації; ефективне згладжування рівномірного шуму.

Недоліки: розмиття контурів; низька ефективність проти імпульсного шуму.

Медіанний фільтр належить до нелінійних методів фільтрації, суть якого полягає в заміні значення пікселя на медіанне значення пікселів, що знаходяться в межах визначеного вікна. Цей підхід відзначається високою ефективністю у видаленні імпульсного шуму, відомого також як шум типу "сіль і перець".

$$I`(x,y) = median\{I(x+i, y+i) | (i,j) \in \Omega\} \quad (1.6)$$

Переваги: ефективне видалення імпульсного шуму; збереження контурів об’єктів.

Недоліки: менш ефективний проти гаусівського або рівномірного шуму; складніший для апаратної реалізації, ніж середній фільтр.

Гаусівський фільтр виконує згладжування за допомогою згортки з ядром, сформованим на основі гаусівського розподілу. Він ефективно згладжує зображення, при цьому краще зберігаючи основні контури порівняно із середнім фільтром.

Ядро формується за формулою:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2}) \quad (1.7)$$

Фільтрація:

$$I`(x,y) = \sum_{(i,j)\in\Omega} G(i,j)\cdot I(x+i,y+j) \quad (1.8)$$

де σ – стандартне відхилення, яке здає ступінь розмиття.

Переваги: добре працює з гаусівським шумом; згладжує без різких втрат контурів.

Недоліки: менш ефективний для імпульсного шуму; потребує обчислення плаваючих чисел.

Традиційні методи фільтрації продовжують демонструвати високу ефективність у базовій обробці зображень, особливо за умови точного знання типу шуму. Втім, їх обмежена здатність адаптуватись зо змішаних шумів і локальних структур створює потребу в застосуванні адаптивних підходів.

1.5 Адаптивні фільтри: принцип дії, класифікація та особливості.

Наукова та прикладна практика містить безліч підходів для фільтрації – від базових методів згладжування до складних адаптивних алгоритмів, які здатні враховувати контекст та структуру зображення. У цьому розділі розглянемо ключові класи методів, їх переваги й обмеження, а також зосередимося на аналізі їхньої реалізації за допомогою Python – універсально інструмента для роботи з візуальними даними.

Одним з перших способів боротьби із шумом стали прості математичні операції. Найкраще відомим є середньозважений (лінійний) фільтр, у якому значення кожного пікселя визначається середнім арифметичним яскравості його сусідів. Такий метод ефективно знижує гаусівський шум, оскільки

зменшення випадковості коливання яскравості. Проте лінійний фільтр має значні недоліки: він не враховує наявність меж і структур об’єкт, що призводить до втрати різкості та розмиття важливих деталей при використанні великих ядер.

Для вирішення зазначеної проблеми було запропоновано використовувати медіанний фільтр, який замінює значення пікселя на медіану значень у локальному вікні. Цей підхід виявляється особливо ефективним у боротьбі з імпульсним шумом, зокрема білим (пересвітленим) або чорним (відсутнім). На відміну від середнього фільтра, медіанний краще зберігає краї та деталі структур, що робить його надзвичайно корисним при обробці зображень, пошкоджених шумом типу «сіль і перець».

Щодо рівномірного шуму, класичні методи демонструють лише помірну ефективність. Такий шум менш агресивний порівняно з імпульсним, але через рівномірний розподіл щільності він суттєво знижує контрастність і деталізацію зображення. Згладжувальні фільтри лише частково усувають цей тип шуму, водночас спричиняючи розмитість.

У мові програмування Python ці методи реалізуються за допомогою таких бібліотек, як OpenCV, SciPy та scikit-image. Наприклад, застосування медіанного фільтра в OpenCV виглядає наступним чином:

```
import cv2
denoised = cv2.medianBlur(image, 3)
```

Цей рядок коду може врятувати знімок, що був зашумлений через недосконаліть сенсора, але він не є універсальним вирішенням усіх проблем.

Одним із значних проривір у сфері фільтрації стала методика частотного аналізу. Її основна ідея базується на припущенні, що шум і корисна інформація мають різний спектральний характер. Зображення за допомогою перетворення Фур’є переходить із просторової області у частотну. У цій області

високочастотні компоненти зазвичай відповідають шумам або дрібним деталям, тоді як низькочастотні частини передають загальну структуру зображення.

Використання ідеальних фільтрів низьких частот (LPF) дозволяє усувати високочастотні шумові коливання. Однак такий підхід має свої недоліки – він може призводити до втрати контрасту, появи ореолів чи інших артефактів після зворотного перетворення.

Більш сучасним рішенням є вайвлет-фільтрація, яка дає змогу аналізувати зображення на різних масштабах. Вайвлети – це функції з локалізованою енергією, які дозволяють розкладати сигнал на часткові елементи з різною деталізацією. Провівши таке декомпозиційне перетворення, можна придушити компоненти, що відповідають за шум, а потім виконати реконструкцію сигналу.

Для Python реалізацію вейвлет-фільтрації забезпечує бібліотека PyWavelets:

```
import pywt
coeffs = pywt.dwt2(image, 'db1')
# подальше приглушення коефіцієнтів та зворотне перетворення
```

Ці методи вирізняються особливою ефективністю проти гаусівського шуму, хоча й не потребують ретельного налаштування. Водночас застосування таких підходів в адаптивних системах сприяє більш точному відновленню зображень без втрати важливих деталей.

Класичні фільтри обробляють все зображення однаково, але реальні умови вимагають більшій гнучкості в підходах. Адаптивна фільтрація полягає в динамічній зміні параметрів залежно від локальних особливостей зони: варіації яскравості, стандартного відхилення, наявності країв та інших характеристик.

Одним із найвідоміших прикладів – адаптивний фільтр Вінера, який спрямований на мінімізацію середньоквадратичної похибки між оригінальним і зашумленим зображенням. Він враховує локальну дисперсію та співвідношення

сигналу до шуму, що дозволяє обережно знижувати шум, зберігаючи при цьому цілісність структури.

Цей підхід добре працює для усунення гаусівського та рівномірного шуму. Однак для боротьби з імпульсним шумом адаптивні фільтри потребують більш складного підходу, який включає виявлення «аномальних» пікселів.

Іншим потужним методом є використання білінійних фільтрів або фільтрів Білатерального зглажування. Вони враховують не тільки просторову відстань між пікселями, але й схожість їхньої яскравості, завдяки чому фільтрація ефективно зглажує лише випадкові шуми, залишаючи краї об’єктів чіткими.

З появою глибокого навчання обробка зображень отримала серйозний розвиток. Нейронні мережі, такі як згорткові автоенкодери та Denoising CNN (DnCNN), можуть вчитися на прикладах зашумлених і чистих пар зображень. Вони не просто видаляють шум, а відтворюють очищену версію, враховуючи контекст.

Для Python доступні фреймворки TensorFlow та PyTorch, які дозволяють реалізовувати ці моделі. Хоча процес навчання потребує значних обчислюваних ресурсів, вже навчені моделі показують чудові результати на гаусівському та рівномірному шумі, а також у складних комбінаціях шумів.

Ці методи вже використовуються для обробки астрономічних знімках, МРТ-томограм, відновлення відео та в інших сферах. У практичному використанні часто неможливо обмежитися лише одним типом шуму. Тому найбільш ефективними є комплексні системи, що поєднують різні підходи: детектують тип шуму, адаптують фільтр до конкретної ділянки зображення та комбінують результати згладжування й реконструкції.

На Python такі системи зручно реалізовувати за допомогою модульної структури:

- Модуль детекції шуму (оцінка дисперсії, гістограма).
- Блок вибору фільтра (медіанний для імпульсного шуму, Вінера для гаусівського).

- Компонент для оптимізації параметрів фільтра.
- Візуалізація та аналіз результатів (з використанням matplotlib, seaborn).

Ця система є не лише гнучкою, але й масштабованою, що дозволяє інтегрувати її у великі проекти, такі як системи комп’ютерного зору чи автоматичної діагностики.

Цифровий світ зображень сповнений складнощів, де шум є не просто дефектом, а серйозною перепоною для точного аналізу. У боротьбі з цією проблемою люди розбирали цілий арсенал методів – від базових фільтрів до передових алгоритмів, заснованих на глибокому навчанні. Однак тільки комплексний підхід, що враховує природу шуму, локальні особливості зображення та ціль обробки, може забезпечити досягнення максимальної якості результатів.

Python зі своїм багатим набором бібліотек та відкритою екосистемою стає ідеальним інструментом для розробки таких рішень. Завдяки поєднанню математичних методів, адаптивних підходів і машинного навчання, він дозволяє створювати потужні системи для обробки зображень, здатні відповідати викликам реального світу.

2 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ

2.1 Визначення функціональних вимог до системи обробки зображень.

Обробка зображення стає критичною складовою не лише наукових і технічних напрямках, а й багатьох прикладних сфер. Її активне застосування спостерігається у медицині, системах безпеки, супутниковій зйомці, промисловості та навіть у повсякденних технологічних рішеннях. Одним із ключових викликів у цій галузі є усунення шуму із зображень, спричиненого різними джерелами. Наявність шуму значно знижує якість візуальних даних, що може призводити до втрати значущої інформації та знижувати ефективність різноманітних задач комп’ютерного зору, таких як сегментація, розпізнавання об’єктів, класифікація чи аналіз.

Система розроблена як модульний застосунок, що містить такі функціональні елементи як:

1. Модульне завантаження зображень (забезпечує підтримку графічних форматів, такі як JPG та PHG.
2. Модуль генерації шумів (дозволяє користувачеві обрати тип шуму).
3. Модуль фільтрації (надає можливість адаптивного вибору фільтрів із заданими параметрами для обробки зображень).
4. Модуль оцінювання якості (виконує автоматичні розрахунки показників якості, таких як PSNR та SSIM).
5. Графічний інтерфейс користувача.
6. Модулю порівняння результатів (дає змогу візуально і наочно аналізувати результати: до/після фільтрації, перегляд графіків і візуалізацій).

Основні завдання дослідження:

1. Виконати аналіз природи та впливу гаусівського, імпульсного й рівномірного шуму на цифрові зображення.

2. Розглянути методи фільтрації шумів і обґрунтувати вибір найбільш ефективних адаптивних фільтрів.

3. Реалізувати алгоритм фільтрації.

4. Створити програму на Python, яка забезпечує:

- Завантаження зображень.
- Генерацію шумів заданих типів.
- Застосування вибраних фільтрів.
- Візуалізацію результатів роботи.

5. Інтегрувати засоби оцінки якості (PSNR, SSIM) для аналізу ефективності фільтрації.

6. Розробити графічний інтерфейс (GUI), орієнтований на зручність користувача.

7. Провести тестування системи на широкому наборі зображень, створивши графіки, таблиці та сформулювати висновки на основі отриманих результатів.

Модульна архітектура цієї системи забезпечує виняткову гнучкість і масштабованість, дозволяючи без зусиль додавати нові функції або вдосконалювати вже наявні. Завдяки інтегрованому підходу, що включає створення шумів, їх обробку, аналіз якості та візуальне порівняння результатів, система стає незамінним інструментом для досліджень у сфері обробки зображень. Водночас вона ефективно працює в прикладних сферах, де критично важлива бездоганна якість візуальних метрик.

2.2 Вибір типів шумів для дослідження та фільтрації.

Незалежно від типу сенсора, чи то цифрова камера, медичний томограф або супутниковий радіометр, будь-яка система фіксації зображень стикається з небажаними флуктуаціями електричного сигналу, квантовими ефектами чи

похибками передавання даних. Потрібні спотворення зазвичай описуються трьома основними видами шуму:

1. Гаусівський шум – це випадкові коливання інтенсивності пікселів, які відповідають нормальному (гаусовому) розподілу.

2. Імпульсивний шум – локальні артефакти у вигляді хаотично розташованих білих пікселів («сіль») або чорних пікселів («перець»), спричинених збоями передавання чи дефектами обладнання.

3. Рівномірний шум – хаотичні відхилення, при яких усі рівні інтенсивності є однаково ймовірними, що створює однорідне «згладжування» деталей без чіткого вираженого піка у розподілі.

Гаусівський шум належить до адитивних шумів, що підпорядковуються нормальному розподілу. Його поява часто пов’язана з тепловими флуктуаціями електроніки сенсорів чи процесами передачі сигналу. Такий шум характеризується тим, що впливає на кожен піксель зображення з певною варіабельністю, що ускладнює визначення уражених зон (рис.2.1). Для боротьби з ним зазвичай застосовують згладжувальні фільтри, які обробляють локальні ділянки, враховуючи середні або зважені значення сусідніх пікселів.

Рисунок 2.1 – Приклад накладання гаусівського шуму на зображення.

Імпульсний шум, відомий також як шум типу «сіль і перець», проявляється у вигляді окремих пікселів із аномально високими чи низькими значеннями, які різко контрастують із сусідньою областю. Незважаючи на

зручність його локалізації, стандартні фільтри для згладжування, такі як середній фільтр, часто виявляються не ефективними, оскільки розмивають зображення без достатнього приглушення імпульсів (рис.2.2). У таких випадках доцільно використовувати медіанний фільтр, який здатний зберігати чіткість контурів і одночасно видаляти ізольовані шуми.

Рисунок 2.2 – Приклад накладання імпульсного шуму.

Рівномірний шум зустрічається рідше, проте стає актуальним у ситуаціях впливу некаліброваних сенсорів чи випадкових перешкод. Для цього притаманний рівномірний розподіл інтенсивностей, що ускладнює його відокремлення від короткого сигналу (2.3). Усунення цього типу шуму потребує більш складних підходів, які комбінують просторову та частотну фільтрацію.

Рисунок 2.3 – Приклад накладання рівномірного шуму.

Кожен із видів шуму по-своєму негативно впливає на якість зображення: гаусівський шум знижує контрастність і розмиває тонкі деталі, імпульсивний створює різкі та помітні артефакти, а рівномірний спричиняє «розмиття» деталей, зменшуючи загальну чіткість. У критичних сферах, таких як медична діагностика, автономне керування си наукове дистанційне зондування, спотворення через шум можуть призводити до серйозних помилок із вагомими наслідками.

Проте існують алгоритми для боротьби з шумами, серед яких різного роду фільтри та адаптивні підходи. Вони дають змогу не лише усувати небажані артефакти, але й зберігати важливі елементи зображення, зокрема його межі, текстури та дрібні деталі. Фільтрація зображень – це не просто процес приглушення шуму. А радше майстерність знаходження балансу між «очищенням» і збереженням ключової інформації.

Основна мета цієї роботи полягає в покращенні якості цифрових зображень при наявності різних типів шуму, таких як гаусівський, імпульсивний та рівномірний, шляхом їх ефективного видалення за допомогою адаптивних фільтрів. Головною ідеєю є визначення та дослідження оптимальних методів фільтрації для кожного типу шуму, порівняння їхньої ефективності за однаковими притеріями та створення комплексної модульної системи, реалізованої за допомогою Python, з графічним інтерфейсом для візуалізації результатів.

2.3 Обґрунтування вибору адаптивних фільтрів.

У рамках цієї роботи пропонується розробка комплексної системи обробки зашумлених зображень на основі адаптивних фільтрів, реалізованої за допомогою програмування на Python. Основною метою є ефективне видалення шуму з одночасним збереженням ключових структурних елементів і деталей зображення. Для досягнення цього завдання важливим є обґрунтований вибір

методів фільтрації, які дозволяють максимально придушити шум при мінімальному спотворенні вихідного зображення.

Спершу необхідно визначити характер і природу проблеми: зашумленість зображень може бути викликана різними типами шуму, кожен із яких вимагає специфічного підходу для його усунення. У межах проєкту зосереджено увагу на трьох найбільших поширених типах шуму: гаусівському, імпульсному та рівномірному. Кожен тип характеризується своїми статистичними властивостями й особливостями розподілу, що визначає специфіку його впливу на піксельні значення зображення.

З огляду на наведені особливості різних типів шуму, стає зрозумілим, що універсального методу, придатного для всіх їх одночасно, не існує. Це зумовлює необхідність створення адаптивної системи фільтрації, яка здатна змінювати свій підхід залежно від характеристик шуму або відповідно до вибору користувача.

Існує безліч методів фільтрації, які широко застосовуються в задачах обробки зображень. Їх можна класифікувати за кількома критеріями: за доменом (просторові чи частотні), за структурою (лінійні, нелінійні, адаптивні) та за механізмом прийняття рішень (фіксовані чи змінні). Однак у контексті розробки програмних систем, здатних ефективно працювати з різними типами шуму в автоматизованому або напівавтоматизованому режимі, найважливішою вимогою є забезпечення адаптивності фільтрації.

Традиційні методи, як-от усереднюючі або гаусівські фільтри чи фільтр згортки з фіксованим ядром, добре справляються з низьким рівнем шуму. Проте вони мають значний недолік: такі підходи розмивають дрібні деталі та нечітко зберігають краї. Медіанний фільтр, натомість, виступає прикладом нелінійного методу, який ефективно усуває імпульсний шум і при цьому не змінює структуру зображення. Водночас його ефективність падає при роботі з гаусівським шумом, де доцільніше використовувати лінійні або частотно-зважені підходи.

З огляду на це, було виправдано включення в систему трьох фільтраційних стратегій, кожна з яких орієнтована на конкретний типу шуму:

Медіанний фільтр використовується для усунення імпульсивного шуму. Цей метод забезпечує стабільну роботу при високій інтенсивності перешкод і дозволяє зберігати контури зображення, що важлива для збереження його структурної чіткості.

$$K(i,j) = median\{I(x,y)|(x,y) \in W(i,j)\} \quad (2.1)$$

де $W(i,j)$ – область фільтрації навколо пікселя (i,j)

Адаптивний фільтр Вінера є найбільш ефективним при гаусівському шумі, оскільки враховує локальні варіації, оцінюючи дисперсію у межах обраного вікна. Він здатний гнучко адаптуватися до текстурованих зон, коригуючи піксельні значення і мінімізуючи втрати деталей.

$$K(i,j) = I(i,j) - \frac{\sigma_n^2}{\sigma_s^2}(I(i,j) - \mu_s) \quad (2.2)$$

де σ_n^2 – дисперсія шуму, σ_s^2 – локальна дисперсія сигналу, μ_s – локальне середнє значення сигналу.

Вейвлет-фільтр, що базується на дискретному вайвлет-перетворені, підходить для обробки змішаного або рівномірного шуму. Завдяки багаторівневій декомпозиції зображення цей метод виділяє високочастотні компоненти та пригнічує їх через порогову обробку.

$$\varphi(t) = \frac{1}{\sqrt{\pi}\,B}\, exp^{-\frac{t^2}{B}} \cdot exp^{j2\pi Ct} \quad (2.3)$$

де B – пропускна здатність, C – центральна частота.

Запропонований підхід є гібридним, оскільки враховує не лише характер шуму, а й структурну складність оброблюваного зображення. Реалізація такої

системи дозволяє користувачеві або самостійно експерементувати із доступним методом фільтрації, або довірити вибір системі, яка автоматично аналізує тип шуму та обирає оптимальний метод.

Серед основних вимог до системи, крім ефективного пригнічення шумів, є збереження ключових характеристик зображення – країв, текстур, тональних переходів. Це особливо актуально для таких завдань, як медична візуалізація, аналіз супутникових даних чи розпізнавання об’єктів, де навіть найменша втрата деталей може призвести до серйозних помилок у трактуванні результатів.

Вибір підходів базувався також на прагматичних міркуваннях щодо продуктивності їх реалізації в середовищі Python. Для інтеграції алгоритмів в інтерфейс і забезпечення швидкої роботи необхідно дотримуватися балансу між ефективністю методів і складністю їх обчислень. Медіанний фільтр, фільтр Вінера та вайвлет-фільтр мають оптимізовану реалізацію у бібліотеках OpenCV, scikit-smage і PyWavelets, що спрощує впровадження та подальшу інтеграцію.

Таким чином, запропонована система побудована за модульним принципом: кожен із методів адаптований до певного типу шуму і може бути викликаний у межах програмного інтерфейсу. Це забезпечує високу гнучкість і зручність використання на практиці.

У процесі розробки комплексної розробки системи для обробки зашумлених зображень важливим є не лише ефективне усунення шуму, а й максимальне збереження інформативних деталей зображення, які можуть втрачатись через надмірно агресивну фільтрацію. З огляду на це, розроблений підхід до фільтрації має задовольняти дві ключові вимоги:

- Гнучкість стосовно різних типів шуму – забезпечення ефективної роботи з різноманітними видами шумів.
- Збереження важливих деталей – увага до збереження текстур, контурів, дрібних елементів і структури.

Адаптивні фільтри – здатні враховувати локальні характеристики зображеннями або заздалегідь підготовленими градаційними версіями зображень. Водночас їхнє застосування є досить обмеженим.

Натомість локальної адаптивні фільтри, зокрема фільтр Вінера, проявляють високу ефективність у задачах, де ключовим є збереження контурів і текстур, особливо якщо йдеться про обробку зображень середнього та високого рівня складності.

У реальних умовах, таких як аналіз фотографій із камер відеоспостереження або супутникових зображень, часто зустрічається шум змішаного типу – наприклад, комбінація імпульсного та гаусіського шуму. За таких обставин найкращі результати забезпечує поетапне або комбіноване застосування кількох фільтрів. Спочатку використовується медіанний фільтр для усунення імпульсних спотворень, а потім – фільтр Вінера або метод вайвлет-декомпозиції для придушення залишкового гаусівського шуму (табл. 2.1).

Особливу увагу заслуговує вейвлет-фільтрація. Ця методика забезпечує не лише збереження структурних елементів зображення, але й високочастотну інформацію, від якої залежить текстура та чіткість. Завдяки пороговій обробці вайвлет-коефіцієнтів стає можливим ефективно відокремити шуми, водночас залишаючи незмінними основні компоненти сигналу.

У підсумку, вибір методу фільтрації базується на таких ключових положеннях:

- Фільтрація має бути гнучкою та враховувати локальний контекст значень пікселів.
- Структура фінального алгоритму повинна бути модульною, що забезпечить можливість комбінування або заміни фільтрів залежно від характерного шуму.
- Реалізація фільтрів має враховувати специфіку програмної платформи, тобто бути оптимальною для використання в Python з застосуванням доступних бібліотек, таких як OpenCV, SciPy, PyWavelets тощо.

- Користувач повинен мати змогу вручну обирати метод фільтрації або покладатися на автоматизований модуль, який визначатиме тип шуму.

Таблиця 2.1 – Порівняльна характеристика обраних фільтрів.

Фільтр	**Тип шуму**	**Адаптивність**	**Збереження контурів**	**Витрати ресурсів**	**Недоліки**
Вінера	Гаусівський	Висока	Добре	Середні	Складність для неоднорідного шуму
Медіанний	Імпульсний	Середня	Відмінно	Низькі	Неефективний для гаусівського шуму
Вайвлет-фільтр	Рівномірний	Висока	Висока	Високі	Потребує точного налаштування

Отже, стратегія фільтрації, що інтегрує медіанні, адаптивні та частотні методи, є найкращим рішенням для створення системи, здатної ефективно працювати з різноманітними та складними ситуаціями шумового зашумлення зображень.

2.4 Вибір мови програмування та бібліотек.

Під час розробки програмної системи для обробки зашумлених зображень, було прийнято низку технологічних рішень, які гарантують високу функціональність, масштабованість і зручність використання. Кожен інструмент або бібліотека обирався з урахуванням технічних особливостей проєкту, практичності реалізації, сумісності компонентів та продуктивності.

Python було обрано основною мовою розробки завдяки її широким можливостям, зручності використання і багатій екосистемі для наукових та технічних задач. Дана мова пропонує численні інструменти для обробки зображень, машинного навчання, аналізу даних і створення графічних

інтерфейсів, що робить Python оптимальним вибором для побудови комплексної системи фільтрації зображень.

Ключові переваги Python:

- Потужна стандартна бібліотека та активна спільнота, яка забезпечує швидке розв’язання проблем і доступу до корисних розширень.
- Мультиплатформенність, яка надає змогу запускати застосунок на Windows, macOC, Linux без значних змін у коді.
- Зручне підключення сторонніх бібліотек через пакетні менеджери, що дозволяє легко додавати новий функціонал.

Таким чином, використання Python дає змогу виконувати весь спектр необхідних обчислень і операцій із зображенням, забезпечуючи при цьому гнучкість розробки, швидке створення прототипів та можливості для подальшого вдосконалення і оптимізації системи.

У процесі розробки системи активно застосовується три ключові бібліотеки для обробки графічних зображень: OpenCV, NumPy PIL та (Pillow). Кожна з них має чітко визначену роль і низку переваг, які спричиняють ефективності роботи проєкту.

OpenCV (Open Source Vision Library) – це потужний інструмент з відкритим кодом, створений для задач комп’ютерного зору та обробки зображень. Бібліотека використовується у різноманітних сферах, зокрема в автоматизації виробничих процесів і технологіях розпізнавання облич.

У рамках проєкту OpenCV застосовується для:

- Завантаження та попередньої обробки зображень;
- Реалізації базових алгоритмів фільтрації;
- Перетворення форматів даних для сумісності з іншими компонентами.

Через оптимізацію на рівні C/C++ і підтримку апаратного прискорення OpenCV забезпечує швидке опрацювання великих обсягів графічних даних у режимі реального часу.

NumPy являється базовим модулем для роботи з багаторівневими масивами та матрицями у Python. Оскільки цифрові зображення фактично представляють собою числові масиви, ця бібліотека слугує критичним елементом проєктної архітектури.

У цьому проекті NumPy дозволяє:

- Створювати, зберігати та змінювати масиви зображень;
- Виконувати розрахунки статистичних параметрів;
- Реалізовувати адаптивні фільтри;
- Проводити векторизовані операції для підвищення продуктивності коду.

Інтеграція можливостей NumPy та OpenCV дає змогу працювати з графічними даними на низькому рівні деталізації й оптимізувати реалізацію фільтрів та інших алгоритмів.

Pillow (вдосконалена версія Python Imaging Library) була обрана для інтеграції графічного ядра з інтерфейсом Tkinter. Вона забезпечує просте перетворення масивів NumPy у формати, які підтримуються елементами GUI Tkinter.

У даному проекті Pillow виконує такі завдання:

- Конвертація форматів зображень;
- Масштабування зображень для подальшого відображення;
- Збереження результатів обробки.

Таким чином, Pillow є сполучною ланкою між обчислювальним ядром системи й візуальною частиною інтерфейсу, що значно спрощує передачу даних до графічного середовища.

Імітація зашумлення зображень є невід’ємною частиною тестування алгоритмів фільтрації. Для цього було реалізовано три основні типи шуму: гаусівський, імпульсний (сіль і перець) та рівномірний. Кожен із цих типів має унікальні характеристики, які різним чином впливають на якість зображення, дозволяючи всебічно оцінити ефективність відповідних методів фільтрації.

Гаусівський шум відтворює вплив електронних перешкод під час передачі зображень. Він може бути створений за допомогою функції numpy.random.normal(), яка додає шум з параметрами середнього значення та стандартного відхилення.

Використовуючи NumPy:

- Генерується матриця шуму, яку додають до вихідного зображення;
- Параметрами регулюється інтенсивність і характер моделювання;
- Забезпечується швидке опрацювання навіть для великих зображень.

Рівномірний шум генерується за допомогою функції numpy.random.normal(), яка створює випадкові значення в певному діапазоні для кожного пікселя. Це сприяє моделюванню фонових перешкод із рівномірним розподілом. Завдяки використанню NumPy:

1. Обчислення виконується з використанням векторизованих операцій, які є значно ефективнішими за циклічні методи;
2. Досягається висока продуктивність навіть у разі роботи з великими обсягами даних.

Імпульсний шум імітує пошкодження, що виникають через збої під час зберігання чи передачі, коли окремі пікселі набувають екстремальних значень. Реалізація алгоритмів NumPy дозволяє:

- З ймовірністю p встановлювати випадкові пікселі на значення 0 або 255;
- Налаштовувати інтенсивність зашумлення через кількість змінених пікселів;
- Масштабувати обчислення для великих зображень без втрати продуктивності.

Така комплексна імітація різних типів шуму забезпечує можливість точного й продуктивного тестування фільтраційних алгоритмів у різних сценаріях.

Система відрізняється інтеграцією трьох високоефективних методів адаптивної фільтрації: медіанного фільтра, фільтра Вінера та вейвлет-

фільтрації. Кожен із них було обрано завдяки їх особливим перевагам у боротьбі з різними типами шумів, що ґрунтується на ретельному аналізі їхньої ефективності.

Медіанний фільтр виступає класичним рішенням для усунення імпульсного шуму. Основна його перевага полягає у здатності ефективно згладжувати різкі локальні зміни без шкоди для контурів зображення.

У проєкті цей алгоритм реалізовано з використанням:

- Функції cv2.medianBlur() з бібліотеки OpenCV, що забезпечує швидку обробку даних;
- Параметра розміру ядра, який налаштовується користувачем для оптимізації фільтрації залежно від інтенсивності шуму.

Завдяки своїй швидкості та особливій ефективності при обробці шуму «солі і перцю», медіанний фільтр демонструє високі технічні показники, підтверджені метриками продуктивності.

Фільтр Вінера спеціалізується на придушенні гаусівського шуму, забезпечуючи мінімізацію середньоквадратичної помилки між отриманим і оригінальним зображенням. Як один із статистичних підходів, він досягає високої якості фільтрації завдяки адаптивним властивостям.

Для реалізації використовується функція «scipy.signal.wiener()» із бібліотеки SciPy, яка:

1. Автоматично враховує локальну варіативність зображення;
2. Не вимагає попереднього визначення точних характеристик шуму;
3. Забезпечує виняткову точність на гладких ділянках.

Метод дозволяє ефективно коригувати як слабко виражений, так і високочастотний шум, що робить його незамінним інструментом у багатьох сценаріях.

Вейвлет-фільтрація базується на розкладі зображення на частотні компоненти із подальшим пороговим очищенням шуму в кожному спектральному каналі. За основу взято бібліотеку PyWavelets (pywt), яка пропонує такі можливості:

- Виконання декомпозиції зображення у вейвлет-базисах;
- Застосування порогових методів обнулення;
- Відновлення зображення з помітно зниженим рівнем шуму.

Цей метод є технічно складнішим у порівнянні з іншими, але виявляє надзвичайну ефективність при комбінованих типах шумів. Завдяки інтеграції з бібліотекою NumPy PyWavelets забезпечує значне спрощення розробки та прискорення обчислювальних процесів.

Кожен зі згаданих методів адаптивної фільтрації демонструє унікальний набір властивостей і переваг, що дозволяє вирішувати широкий спектр із видалення шуму.

2.5 Архітектура програмної системи: модулі, взаємодія компонентів.

Проектування архітектури програмного забезпечення є критично важливим етапом у розробці складних систем, забезпечуючи узгодженість, масштабованість і ефективність виконання функцій. У випадку задач, пов’язаних з обробкою зашумлених зображень, архітектура повинна підтримувати модульність, гнучкість налаштувань, високу продуктивність при роботі з графічними даними і зручність для користувача.

Ключові вимоги до архітектури системи обробки зображень:

1. Модульність: кожен компонент (обробка зображень, фільтрація, завантаження/збереження, інтерфейс користувача) має бути незалежним, що спрощує оновлення і тестування частин системи.
2. Розширюваність: можливість інтеграції нових фільтрів або моделей шуму без зміни основної логіки системи.
3. Кросплатформеність: система повинна функціонувати на різних операційних системах, таких як Windows, Linux і macOS.

4. Інтерактивність: надання користувачам можливості змінювати параметри фільтрації і миттєво бачити результати в режимі реального часу.

5. Продуктивність: впровадження оптимізованих алгоритмів і структур даних для ефективної роботи з великими зображеннями.

Усвідомлюючи ці вимоги, було вирішено застосувати архітектурний підхід MVC (Model-View-Controller) як основну парадигму для побудови системи.

Шаблон Model-View-Controller забезпечує чітке розподілення логіки даних (Model), візуального представлення (View) та взаємодії з користувачем (Controller). Такий підхід дозволяє:

- Ізолювати обчислювальну частину (фільтрація, обробка) від інтерфейсу.
- Спростити налагодження та тестування окремих модулів.
- Забезпечити підтримку багатьох фільтрів без порушення логіки відображення.

Застосування MVC в Python реалізовано через модульну організацію каталогів і файлів. Основні компоненти:

- Model – містить логіку фільтрації, генерації шумів, розрахунок метрик.
- View – відображає графічний інтерфейс користувача.
- Controller – обробляє дії користувача, ініціює виконання фільтрації, відповідає за маршрутизацію даних.

Система складається з основних складових елементів, кожен із яких виконує специфічні функції:

1. Модуль для роботи з файлами.

Відповідає за завантаження зображень із локального сховища, їхнє збереження після обробки, а також формування зображень для подальшої роботи.

2. Модуль для створення шумів.

Забезпечує генерацію шумів таких типів: гаусівський, імпульсний і рівномірний. Користувач може налаштовувати параметри шуму в режимі реального часу через графічний інтерфейс.

3. Модуль фільтрування.

Містять реалізацію таких фільтрів: медіанний фільтр, фільтр Вінера та вейвлет-фільтр. Є можливість гнучкого розширення функціоналу за допомогою динамічного підключення нових фільтрів, реалізованого на основі патерну «Стратегія».

4. Модуль обробки зображень.

Виконує ключові операції: перетворення зображень у відтінки сірого, зміну розмірів, нормалізацію даних, а також відновлення кольоровості після застосування алгоритмів обробки.

5. Модуль оцінки якості обробки.

Обчислює метрики якості, такі як PSNG, SSIM, та MAE. Результати автоматичного передаються контролеру та відображаються для ознайомлення користувачу.

6. Графічний інтерфейс користувача.

Реалізований із використанням фреймворку KivyMD. Забезпечує зручну та інтуїтивну взаємодію з системою, дозволяючи переглядати результати у форматі "до/після", а також налаштовувати параметри роботи алгоритмів.

У реалізації використовується наступні бібліотеки для мови програмування Python:

- OpenCV – для роботи із зображеннями (завантаження, фільтрація, перетворення форматів).
- NumPy – ефективна обробка масивів пікселів.
- scikit-image – допоміжна бібліотека для фільтрації, додавання шумів та метрик якості.
- PyWavelets – бібліотека для вейвлет-фільтрації.
- Kivy + KivyMD – для створення кросплатформеного GUI.

Було розглянуто та обґрунтовано архітектуру програмної системи для обробки зашумлених зображень, побудовану на основі принципів модульності, масштабованості та зручності використання. Використання структури MVC забезпечує чітке розділення функції між компонентами, що спрощує підтримку та подальше розширення системи. Впровадження шаблонів проєктування, таких як Strategy і Factory Method, надає системі гнучкість у додаванні нових алгоритмів фільтрації та типів шумів. Завдяки інтеграції бібліотек OpenCV, scikit-image та KivyMD реалізовано ефективну, сучасну і кросплатформену систему, здатну якісно виконувати обробку зображень у різноманітних умовах.

2.6 Формалізація критеріїв якості фільтрації (PSNR, SSIM)

Оцінка якості обробки зашумлених зображень є надзвичайно важливим етапом у процесі аналізу ефективності роботи методів фільтрації та покращення зображень. Вона дозволяє визначити, наскільки добре методи очищення справляються зі своєю задачею та наскільки близьким до початкового вигляду стає зображення після видалення шумів. Застосування формалізованих та обґрунтованих метрик для оцінювання результатів такого відновлення є ключем до об’єктивного аналізу параметрів якості. Такі метрики забезпечують можливість порівняння різних методів або алгоритмів, використовуючи числові показники, що дозволяють зменшити суб'єктивність у прийнятті рішень щодо ефективності відновлення.

PSNR – метрика, яка ґрунтується на використанні середньоквадратичної похибки (MSE) для визначення рівня відхилення зміненого зображення від оригінального, обчислюючи відношення максимально можливої потужності сигналу до потужності шуму. Ця метрика представляє результат у логарифмічній шкалі, що дозволяє більш адекватно оцінювати широкий діапазон значень.

$$PSNR = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE}) , \quad (2.4)$$

де M, N – розміри зображення.

Вона не тільки надає змогу виміряти рівень шуму в обробленому зображенні, але й дозволяє встановити ступінь його відповідності до оригіналу на основі точних числових підрахунків.

SSIM – яка використовується для оцінки змін у структурі зображення з точки зору того, як їх сприймає людський зір. Вона побудована на принципі, що людський зоровий апарат є чутливим до таких аспектів, як композиція світла та деталей, і базується на трьох ключових компонентах: яскравості, контрастності та структурних особливостях.

Формула SSIM для вікон *x* та *y* зображень:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} , \quad (2.5)$$

де $\mu_{x,}\mu_y$ – середні значення вікон; σ_x^2, σ_y^2 – дисперсія; σ_{xy} – коваріація, C_1, C_2 – стабілізуючі константи.

Ця метрика орієнтована на те, щоб максимально точно відобразити суб'єктивне сприйняття змін у візуальній інформації, забезпечуючи більш природний підхід до оцінки якості зображення порівняно з традиційними методами.

В межах впровадженої системи, всі три метрики застосовуються для оцінки ефективності класичних і адаптивних фільтрів. Такий комплексний підхід надає декілька важливих переваг:

- забезпечення об'єктивної оцінки результатів, що базується на точних алгоритмічних розрахунках;
- надання перцептивної оцінки, яка наближена до реального людського сприйняття і дозволяє краще розуміти практичну придатність фільтрів;

- створення можливостей для формування детальних таблиць ефективності, які можна використовувати для автоматизованого порівняння параметрів і продуктивності різних типів фільтрів.

Об’єднане використання метрик PSNR і SSIM дозволяє компенсувати недоліки кожної з них, забезпечуючи більш повний аналіз. PSNR пропонує об’єктивну оцінку втрат сигналу за допомогою математичних обчислень, тоді як SSIM фокусується на структурних особливостях зображення, враховуючи особливості людського сприйняття. Такий комплексний підхід дає можливість оцінювати як технічну якість процесу фільтрації, так і його візуальне сприйняття. Це має вирішальне значення при визначенні найбільш ефективних методів обробки зображень із шумами.

Це дозволяє суттєво полегшити аналіз і вибір оптимальних рішень, враховуючи як технічні аспекти, так і їхнє сприйняття користувачами.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ТА ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Реалізація інтерфейсу завантаження та попереднього перегляду зображень.

В ході розробки модулів для обробки та фільтрації зображень була створена гнучка й легко розширювана система, яка поєднує передові алгоритми додавання шумів і адаптивного фільтрування з зручним і зрозумілим інтерфейсом. Кожен модуль має чітко визначену спеціалізацію: від генерації зашумлених зображень до їхнього ефективного очищення за допомогою таких методів, як медіанний фільтр, вейвлет-фільтр і фільтр Вінера. Використання надійних бібліотек, зокрема NumPy, OpenCV, SciPy та PyWavelets, гарантує високу точність обробки та стабільну продуктивність системи.

Усі функції адаптовані для використання через графічний інтерфейс. Наприклад, користувач може обрати тип шуму, натиснути кнопку та переглянути результат фільтрації. Це забезпечує взаємодію між модулями обробки зображень та інтерфейсом, створюючи інтерактивну і зручну для користувача систему.

Об’єднання цих модулів у єдине програмне рішення на основі Tkinter дозволило створити повноцінний інструмент для оцінки ефективності методів фільтрації зображень. Користувач може не лише візуально спостерігати зміни, але й отримувати чисельну оцінку якості зображень за допомогою метрик PSNR і SSIM. Такий підхід робить систему однаково зручною як для наукових досліджень, так і для вирішення практичних задач у сферах комп’ютерного зору та цифрової обробки зображень.

Одним із ключових аспектів комплексної системи є її доступність для кінцевого користувача. Для забезпечення зручності взаємодії у проєкті розроблено графічний інтерфейс користувача (GUI) на основі Tkinter – стандартної бібліотеки Python для створення віконних застосунків.

Tkinter має низку переваг, які зумовили його використання для побудови графічного інтерфейсу:

- Інтегрованість із Python. Дана бібліотека є частиною стандартного пакета, що усуває необіхдність у додатковому встановленні та сприяє підвищенню портативності застосунку;
- Легкість у використанні. Інтуїтивний синтаксис і доступна система геометричних менеджерів дозволяють швидко й ефективно створювати елементи управління;
- Інтеграція з PIL. Підтримка роботи з об’єктами зображень спрощує відображення, оновлення та обробку візуального контенту;
- Подієво-орієнтоване програмування. Забезпечує можливість оперативного реагування на дії користувача в режимі реального часу, що дозволяє завантажувати зображення, вибрати тип шуму чи фільтра на переглядати результати;
- Ефективність у споживанні ресурсів. Порівняно з більш важкими GUI-фреймворки Tkinter не навантажує систему, демонструючи високу швидку дію навіть на застарілих пристроях.

Графічний інтерфейс складається з декількох ключових елементів (рис.3.1):

1. Завантаження зображення: кнопка вибору файлів та вікно для завантаження вхідного зображення.
2. Додавання шуму: набір радіокнопок для вибору типу шуму.
3. Вибір фільтра: можливість обрати метод фільтрації.
4. Відображення результатів:

- Оригінальне зображення;
- Зображення із доданим шумом;
- Результат після фільтрації.

5. Обчислення метрик: автоматичний розрахунок PSNR та SSIM із виведенням отриманих значень.

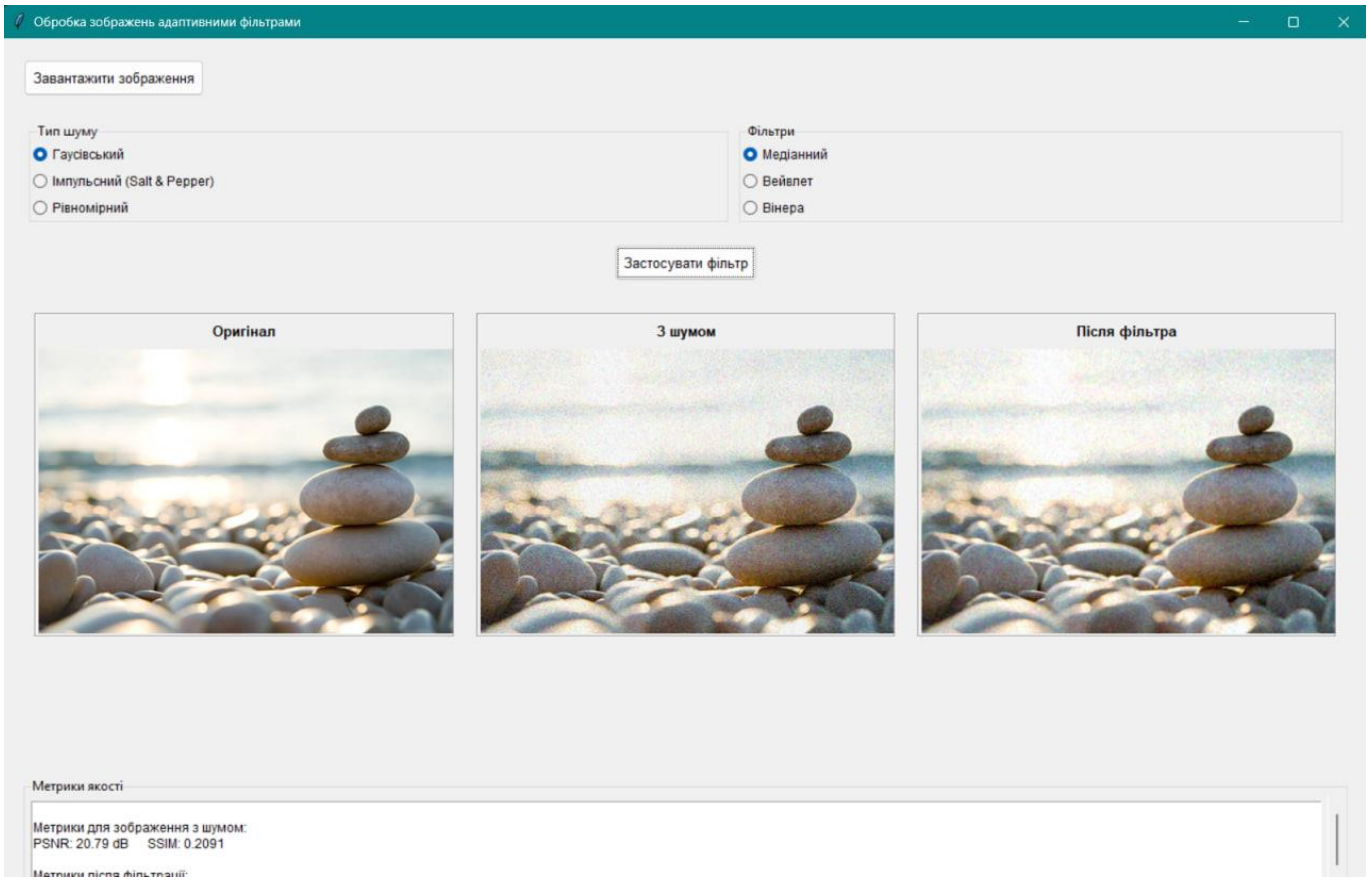

Рисунок 3.1 – Графічний інтерфейс додатка.

Завдяки модульному підходу інтерфейс є гнучким – його легко змінювати, додаючи нові фільтри або типи шумів. Це підтверджує перевагу використання Tkinter як основи розробки.

3.2 Модуль додавання штучного шуму для тестування фільтрів.

У процесі створення та аналізу ефективності алгоритмів обробки зображень критично важливим є забезпечення можливості їх тестування в умовах, максимально наближених до реальних сценаріїв використання. Одним із найбільш значущих факторів, які необхідно враховувати, є здатність алгоритму справлятися з різними видами шуму, що часто виникають під час роботи із зображеннями. Для вирішення цього завдання передбачено спеціальний модуль, який надає широкий спектр функцій, що дозволяють

генерувати штучний шум різних типів. Цей підхід дає змогу моделювати різноманітні види погіршень якості зображення, створюючи умови для ретельного тестування алгоритмів шумозаглушення. Завдяки такій функціональності стає можливим більш глибокий аналіз ефективності застосовуваних фільтрів і вибір оптимальних рішень для роботи з реальними даними.

Гаусівський шум характеризується випадковими варіаціями інтенсивності пікселів, що слідують нормальному розподілу:

```
def add_gaussian_noise(image, mean=0, var=0.01):
    sigma = var ** 0.5
    gaussian = np.random.normal(mean, sigma, image.shape)
    noisy_image = image + gaussian * 255
    return np.clip(noisy_image, 0, 255).astype(np.uint8)
```

Такий підхід імітує шуми, що виникають під час передачі сигналу чи зйомки при недостатньому освітленні.

Імпульсний шум проявляється у випадковій зміні деяких пікселів до крайніх значень (0 або 255):

```
def add_salt_and_pepper_noise(image, amount=0.05):
    noisy_image = np.copy(image)
    num_salt = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i, int(num_salt)) for i in image.shape]
    noisy_image[tuple(coords)] = 255

    num_pepper = np.ceil(amount * image.size * 0.5)
    coords = [np.random.randint(0, i, int(num_pepper)) for i in image.shape]
    noisy_image[tuple(coords)] = 0
            return noisy_image
```

Такий вид шуму зазвичай моделює пошкодження, викликані, наприклад, проблемами з’єднання або помилками під час зберігання зображення.

Рівномірний шум утворюється шляхом додавання випадкових значень в межах визначеного діапазону:

```
def add_uniform_noise(image, low=-30, high=30):
    noise = np.random.uniform(low, high, image.shape)
    noisy_image = image + noise
            return np.clip(noisy_image, 0, 255).astype(np.uint8)
```

Він ефективний для моделювання шумів, що виникають через датчики або зовнішнє середовище, де ймовірність однаково розподілена у визначеному діапазоні (рис. 3.2).

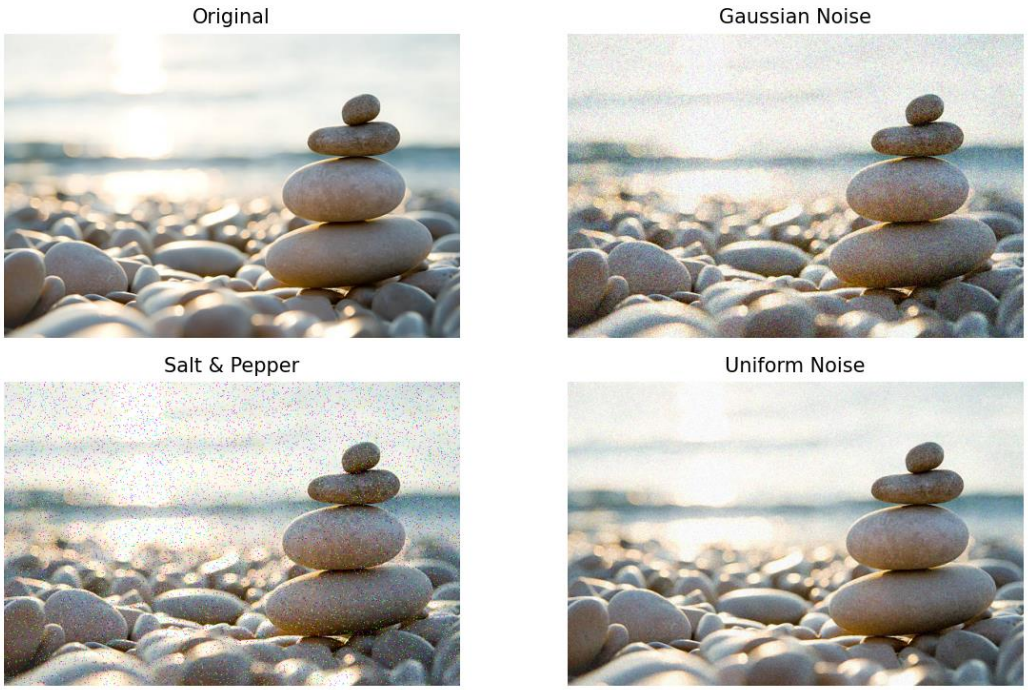

Рисунок 3.2 – Зашумлені зображення обраними фільтрами.

Цей модуль пропонує потужний інструментарій для генерації зашумлених зображень, що відіграє ключову роль у комплексному тестуванні алгоритмів фільтрації. Завдяки моделюванню різноманітних типів шуму

розробники можуть оцінювати ефективність своїх методів та вдосконалювати їх для забезпечення максимальної точності в умовах реальних даних. Застосування цих функціональних можливостей дозволяє детально аналізувати продуктивність фільтрів і їх здатність відновлювати початковий сигнал.

3.3 Реалізація адаптивних фільтрів.

Ефективне зменшення шуму є визначальним етапом у процесі обробки зображень, особливо коли необхідно зберегти ключові деталі та структури. На відміну від статичних фільтрів, адаптивні фільтри підлаштовують свою роботу відповідно до локальних особливостей зображення. Це дозволяє їм більш ефективно усувати різноманітні типи шуму, мінімізуючи водночас втрати важливої інформації. Гнучкість таких фільтрів виявляється у здатності автоматично коригувати їхні параметри, наприклад, розміри вікон або коефіцієнти, для кожної конкретної ділянки зображення. Вони реагують на зміни інтенсивності, текстури чи наявність шумів, що робить цей підхід значно універсальнішим і результативнішим порівняно зі статичними методами, які застосовують однакові операції до всього зображення без врахування його локальних властивостей.

Медіанний фільтр ефективний проти імпульсного шуму, оскільки замінює значення пікселя на медіанне значення серед сусідніх пікселів:

```
def median_filter(image, kernel_size=3):
        return cv2.medianBlur(image, kernel_size)
```

Цей метод забезпечує менше розмивання контурів порівняно із середньозваженим фільтром, що дозволяє зберегти більше людей.

Вейвлет-фільтр використовується для видалення високочастотного шуму через декомпозицію зображення на частотні складові:

```
def wavelet_filter(image, wavelet='db1', level=1):
    coeffs = pywt.wavedec2(image, wavelet, level=level)
    coeffs_filtered = list(coeffs)
    coeffs_filtered[1:] = [(pywt.threshold(c, 20, mode='soft') for c in level) for level in coeffs[1:]]
        return np.clip(pywt.waverec2(coeffs_filtered, wavelet), 0, 255).astype(np.uint8)
```

Підхід із м’яким порогуванням дозволяє зменшувати фоновий шум, зберігаючи сигнали з високою енергією.

Адаптивний фільтр Вінера адаптується до локальної статистики зображення, наприклад дисперсії:

```
def wiener_filter(image, kernel_size=5):
        return wiener(image, (kernel_size, kernel_size))
```

Він особливо ефективний для фільтрації гаусівського шуму, враховуючи локальні варіації яскравості.

Адаптивні фільтри є потужним інструментом для покращення якості зображень, адже вони дозволяють знаходити оптимальний баланс між зменшенням шуму та збереженням важливих деталей. Ці фільтри виразно демонструють перевагу підходів, що враховують локальні характеристики зображення, забезпечуючи більш точну та ефективну обробку. Вибір конкретного методу залежить від характеру присутнього шуму та вимог до кінцевого результату. Їх впровадження дає змогу гнучко працювати з багатьма сценаріями, підвищуючи ефективність систем обробки зображень і дозволяючи досягати високоякісних результатів навіть у роботі з пошкодженими даними. Подальший розвиток цієї галузі може включати комбінування існуючих методів або створення нових адаптивних підходів, які враховуватимуть ще складніші моделі шуму та факторів деградації.

3.4 Візуалізація результатів фільтрації: «до» та «після».

Візуалізація результатів, отриманих після процесу фільтрації, є надзвичайно важливим етапом, оскільки саме вона дає змогу комплексно оцінити ефективність обраних алгоритмів для очищення зображень. Хоча числові метрики, такі як PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index Measure), надають об'єктивну інформацію про якість обробки зображення, значну роль відіграє також візуальне порівняння. Це порівняння дозволяє оцінити важливі аспекти, як-от збереження дрібних деталей, чіткість контурів та ступінь видалення присутнього шуму.

У межах даної системи візуалізація реалізована засобами графічного інтерфейсу користувача, побудованого на основі інструментів бібліотеки Tkinter. Для безпосереднього відображення окремих зображень застосовано можливості бібліотеки Matplotlib, яка забезпечує зручний і гнучкий підхід до роботи з графіками та візуальними даними. Таким чином, користувач має змогу не лише аналізувати числові показники, але й швидко отримувати наочну оцінку роботи алгоритмів, що особливо важливо при налаштуванні та вдосконаленні системи очищення.

Алгоритм починається із завантаження вихідного зображення, яке доповнюється штучним шумом (рис. 3.3). Далі застосовується один із доступних фільтрів, таких як медіанний, гаусівський, фільтр Вінера або вейвлет. Після цього результати демонструються у форматі порівняння «до/після»:

- Оригінальне (без шуму);
- Зашумлене;
- Відфільтроване.

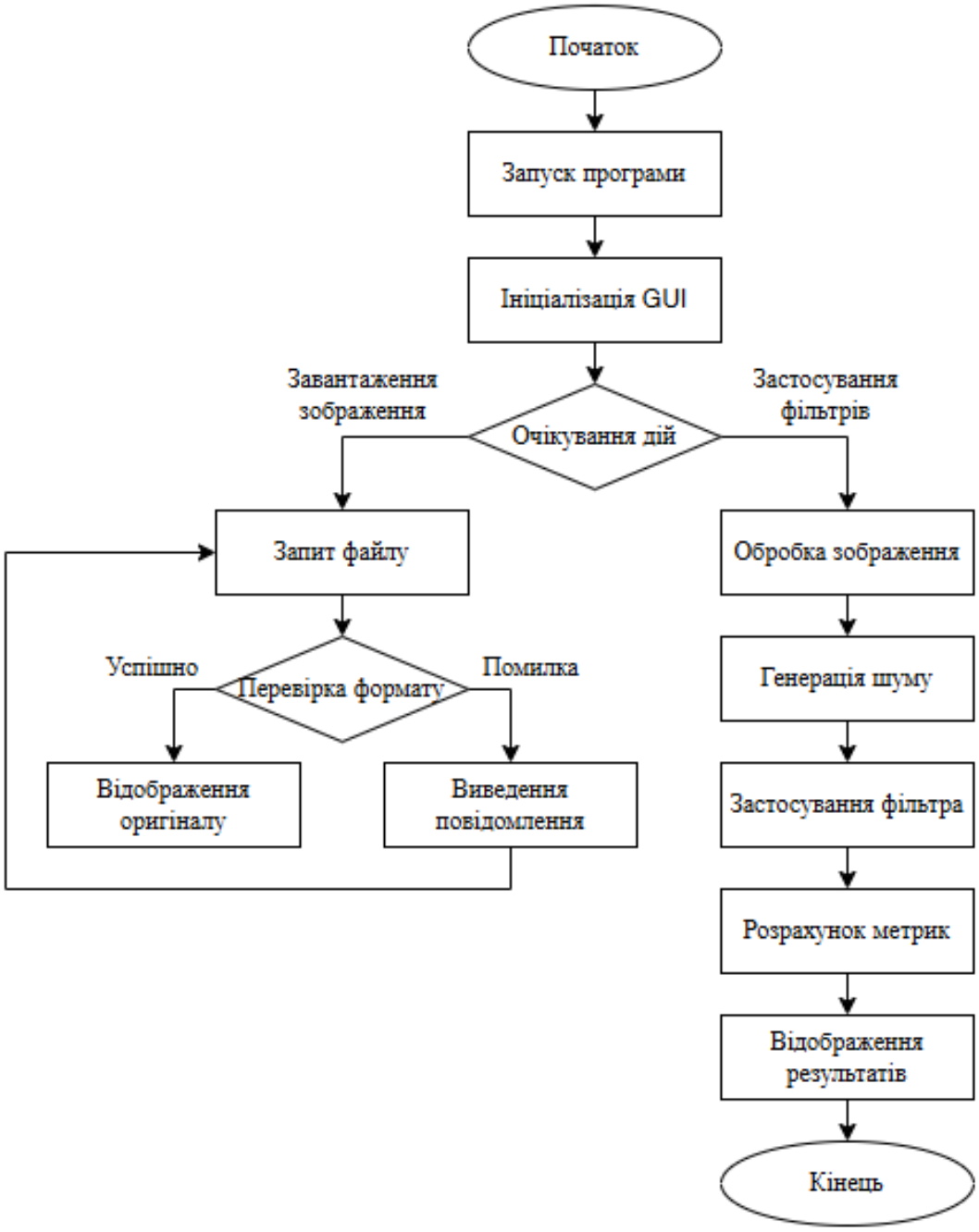

Рисунок 3.3 – Основний алгоритм роботи.

Даний підхід дозволяє оцінювати якість фільтрації не лише за допомогою числових показників, а й наглядно, на основі візуального аналізу (рис. 3.4 – 3.12).

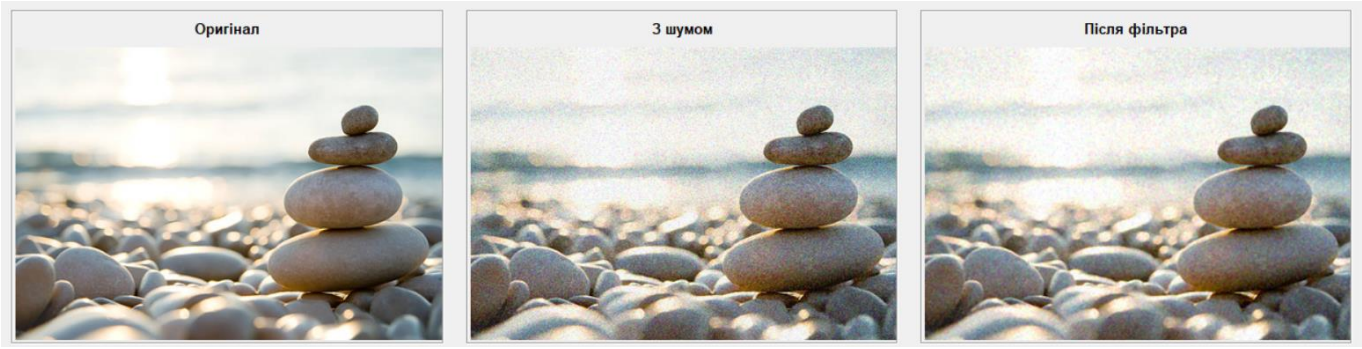

Рисунок 3.4 – Результат медіанної фільтрації зображення з гаусівським шумом.

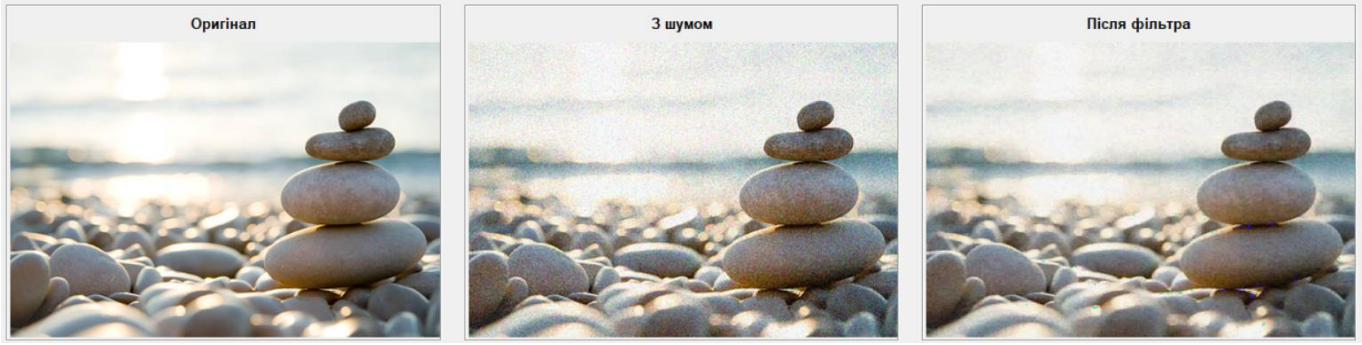

Рисунок 3.5 – Результат вейвлет-фільтрації з гаусівським шумом.

Рисунок 3.6 – Результат фільтрації Вінера з гаусівським шумом.

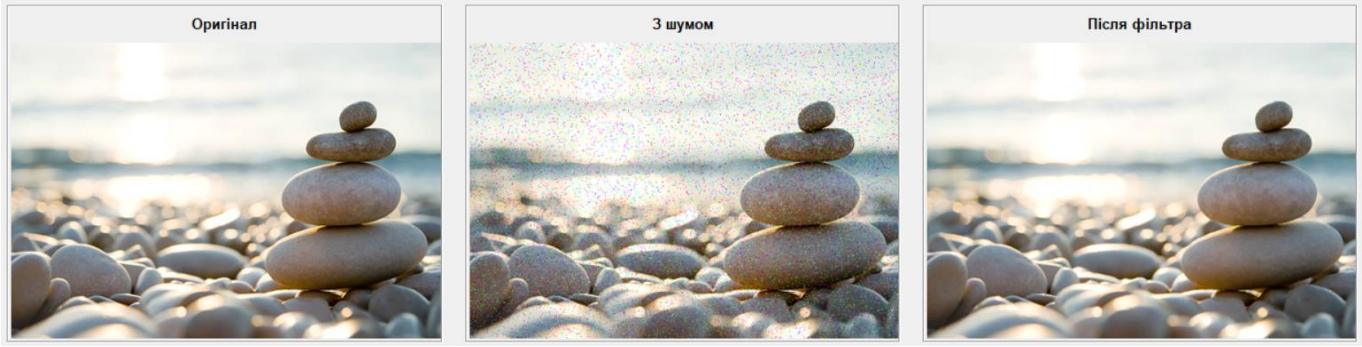

Рисунок 3.7 – Результат медіанної фільтрації з імпульсним шумом.

Рисунок 3.8 – Результат вейвлет-фільтрації з імпульсним шумом.

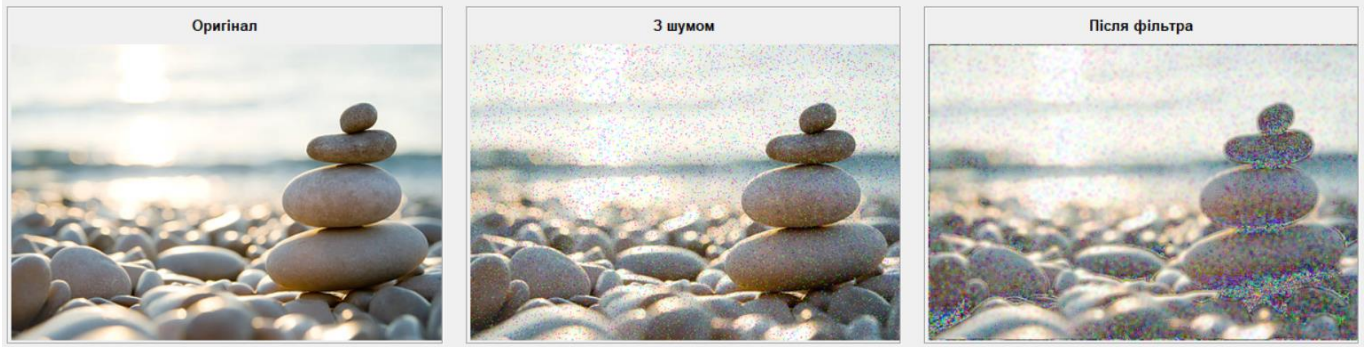

Рисунок 3.9 – Результат фільтрації Вінера з імпульсним шумом.

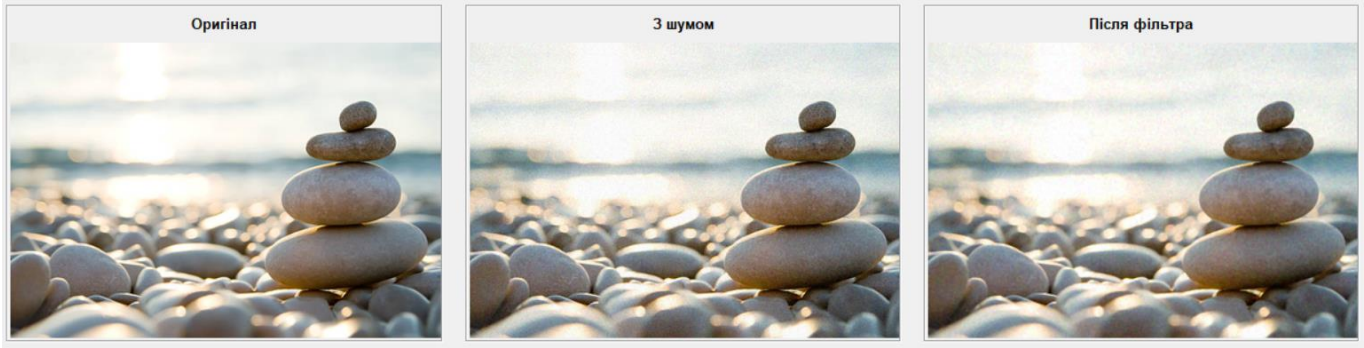

Рисунок 3.10 – Результат медіанної фільтрації з рівномірним шумом.

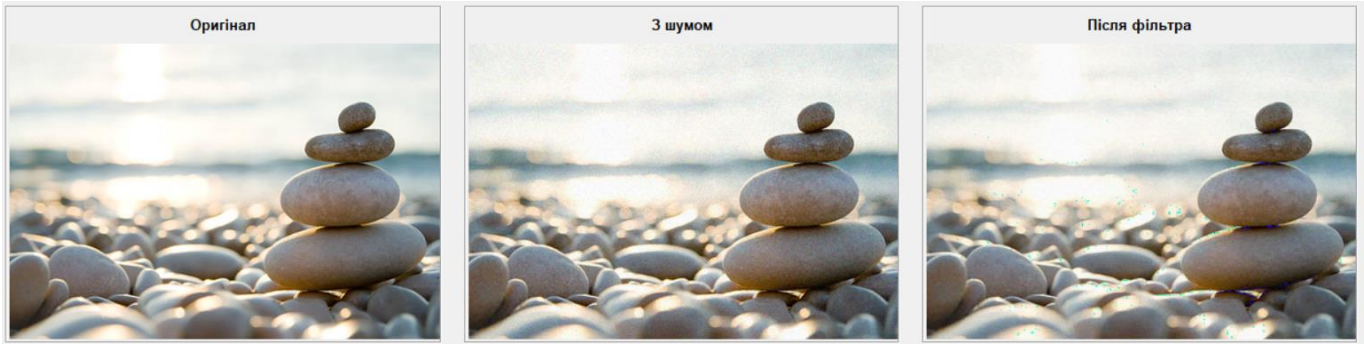

Рисунок 3.11 – Результат вейвлет-фільтрації з рівномірним шумом.

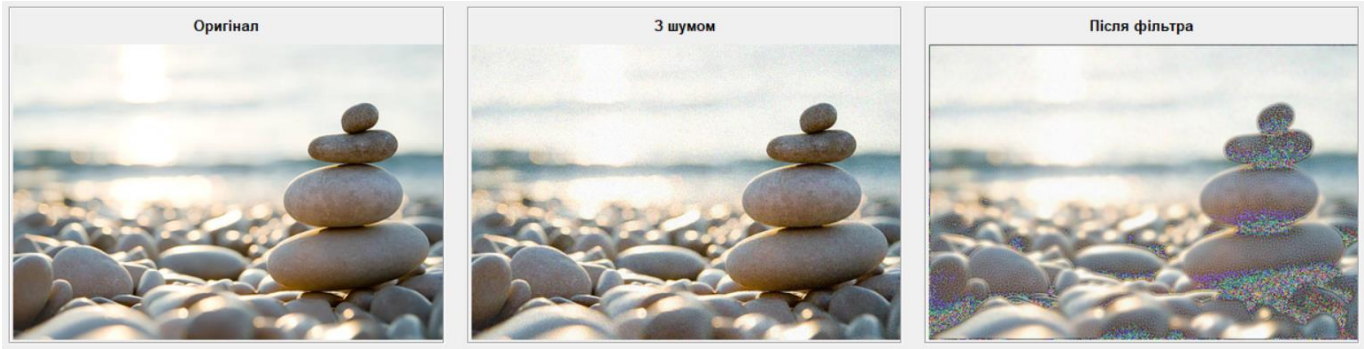

Рисунок 3.12 – Результат фільтрації Вінера з рівномірним шумом.

Візуальне порівняння зображень до та після застосування фільтрації виконує надзвичайно важливу роль у процесі аналізу, адже дозволяє:

- з легкістю та оперативністю оцінити рівень ефективності використаного фільтра, помічаючи як позитивні зміни, так і потенційні недоліки;
- виявити можливі проблеми зі зменшенням деталізації чи появою небажаного розмиття, які можуть суттєво вплинути на якість обробленого зображення;
- перевірити й підтвердити результат обробки, впевнившись, що він відповідає очікуваній структурі зображення та не порушує його ключових характеристик.

У поєднанні з використанням числових метрик, таких як PSNR (пікова відношення сигналу до шуму) та SSIM (індекс структурної подібності), така візуальна оцінка стає одним із найбільш надійних та ефективних методів комплексного аналізу якості, коли йдеться про процеси адаптивної фільтрації даних.

3.5 Оцінювання якості відновлення зображень з використанням метрик.

Для успішної реалізації системи обробки зображень, спотворених шумами, недостатньо лише теоретично обґрунтувати методи фільтрації – необхідно також ретельно перевірити їх на ретельних прикладах. Експериментальне дослідження дає змогу оцінити ефективність різноманітних фільтрів за заданих умов, специфіку роботи адаптивних підходів залежно від типу шуму. У ході цієї роботи було проведено всебічне тестування трьох видів шуму – гаусівського, імпульсного та рівномірного. У якості тестового базового зображення використано еталонне зображення, що дозволило комплексно оцінити продуктивність трьох фільтрів: медіанного, вейвлет-фільтра та адаптивного фільтра Вінера.

Процес тестування базувався на використані стандартизованої послідовності операцій:

1. Завантаження еталонного зображення.
2. Додавання до зображення шуму певного типу.
3. Застосування одного із трьох фільтрів для обробки зашумленого зображення.
4. Розрахунок показників якості зображення до та після застосування фільтрації:

- PSNR (відношення сигналу до шуму за піковим значенням);
- SSIM (індекс структурної подібності).

Для забезпечення максимальної наочності результати тестування представлені у вигляді метрик у табличні формі, доповнені візуалізацією отриманих зображень для кожного окремого випадку (рис. 3.13).

Рисунок 3.13 – Оригінальне зображення.

Гаусівський шум характеризується змінами яскравості пікселів відповідно до нормального розподілу. Цей тип шуму широко поширений у практичних задачах комп’ютерного зору, зокрема у галузях медичної візуалізації, обробки супутникових знімків, медичних обстежень та інших схожих напрямках. Проведений аналіз методів фільтрації показів, що найбільш ефективним виявився вейвлет-фільтр, який забезпечив найкращі значення як PSNR, так і SSIM. Помітні покращення були також досягнуті за допомогою медіанного фільтра. Натомість результати фільтра Вінера виявились неоднозначними: під час його застосування спостерігалось підвищення SSIM, але водночас зниження PSNR, що свідчить про певну втрату сигналу (табл. 3.1).

Таблиця 3.1 – Результати метрик для гаусівського шуму.

Фільтр	PSNR (до)	PSNR (після)	Δ PSNR	SSIM (до)	SSIM (після)
Медіанний	20,79 dB	27,31 dB	+6,52	0,2093	0,5964
Вейвлет-фільтр	20,78 dB	28,16 dB	+7,37	0,2086	0,7775
Фільтр Вінера	20,78 dB	14,44 dB	-5,34	0,2092	0,5314

Імпульсний шум, ще відомий як «сіль і перець», особливо ефективно усувається за допомогою медіанного фільтра. Цей метод має здатність структурно виявляти та видаляти окремі пікселі з аномальною яскравістю. У результаті застосування медіанного фільтра вдалося досягнути вражаючого поліпшення: підвищення PSNR більш ніж на 18 dB та SSIM майже до 0,96, що практично відновлює початкову структуру зображення. Водночас вейвлет-

фільтр Вінера демонструють значно гірші результати, подекуди навіть негативний ефект, що підкреслює їхню неефективність такого виду шуму (табл. 3.2).

Таблиця 3.2 – Результати метрик для імпульсного шуму.

Фільтр	PSNR (до)	PSNR (після)	Δ PSNR	SSIM (до)	SSIM (після)
Медіанний	17,47 dB	36,33 dB	+18,85	0,7066	0,9589
Вейвлет-фільтр	17,50 dB	16,62 dB	-0,89	0,7075	0,6554
Фільтр Вінера	17,47 dB	17,43 dB	-0,05	0,7058	0,5208

Рівномірний шум має дещо інший характер: випадкові значення пікселів розподіляються рівномірно по всьому діапазону. Такий вид шуму суттєво знижує локальну різкість, тому основне завдання фільтрації полягає у відновленні текстур і контурів. Примітно, що в цьому випадку найбільш ефективно проявив себе медіанний фільтр, хоча результати отримані за допомогою вейвлет-фільтра, були практично такими ж. Фільтр Вінера, у свою чергу, показав неоднозначні результати: покращився показник SSIM, але PSNR зменшився, що може свідчити про втрату дрібних деталей зображення (табл. 3.3).

Таблиця 3.3 – Результати метрик для рівномірного шуму.

Фільтр	PSNR (до)	PSNR (після)	Δ PSNR	SSIM (до)	SSIM (після)
Медіанний	27,56 dB	31,69 dB	+4,13	0,4763	0,7874
Вейвлет-фільтр	27,56 dB	28,47 dB	+0,91	0,4763	0,8537
Фільтр Вінера	27,56 dB	17,00 dB	-10,57	0,4763	0,5984

Медіанний фільтр зарекомендував себе як надзвичайно ефективний інструмент для усунення імпульсного шуму, успішно справляючись із завданнями його пригнічення. Крім того, його стабільна робота при обробці інших видів шумів робить цей метод надзвичайно універсальним і придатним

для вирішення широкого спектра задач у цифровій обробці сигналів та зображень

Вейвлет-фільтр своєю чергою демонструє найвищий рівень структурної подібності (SSIM) серед проаналізованих методів, що забезпечує високоякісне відновлення зображень у більшості випадків. Особливо помітно його перевагу можна спостерігати під час боротьби з гаусівським шумом чи рівномірно розподіленим шумом, де він показує найкращі результати в контексті збереження структури об’єктів.

Фільтр Вінера виявився менш стабільним за показником пікової відносини сигналу до шуму (PSNR), де в багатьох випадках його значення опинялися на середньому або навіть нижчому рівні. Проте в окремих ситуаціях цей фільтр здатний забезпечувати прийнятний рівень структурної подібності (SSIM), що може робити його доречним вибором для вузькоспеціалізованих умов застосування.

Медіанний фільтр являє собою універсальний та ефективний інструмент для усунення імпульсного шуму та водночас забезпечує стабільну роботу з іншими типами шумів. Вейвлет-фільтр відзначається здатністю якісно відновлювати зображення, демонструючи найвищий рівень структурної подібності (SSIM), особливо успішно справляючись із гаусівським та рівномірно розподіленим шумом. Фільтр Вінера, хоча і має менш стабільні показники PSNR, залишається корисним для виконання специфічних завдань завдяки можливості досягати прийнятного рівня SSIM у певних умовах.

4 ТЕСТУВАННЯ, ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

4.1 Порівняння ефективності фільтрів за обраними метриками.

Результати фільтрації зашумлених зображень із використанням адаптивних методів вимагають ґрунтовного аналізу, який допоможе визначити ефективність кожного підходу залежно від характеру накладеного шуму. У цьому контексті використовувались об’єктивні метрики оцінки якості зображень, такі як PSNR (пік сигналу до шуму) та SSIM (індекс) структурної подібності), які дозволяють чисельно оцінювати ступінь спотворення й рівень збереження структурних характеристик після фільтрації. Окрім того, додатково враховувались візуальні характеристики, щоб оцінити, наскільки кінцеве зображення виглядає природним для сприйняття оком людини. Такий підхід дозволив комплексно оцінити ефективність застосованих методів фільтрації.

Дані були отримані під час роботи з трьома основними типами шуму: гаусівським, імпульсним і рівномірним. Для кожного з них протестовано три різні методи фільтрації: медіанний фільтр, вейвлет-фільтр і фільтр Вінера. Це дало змогу виявити переваги та обмеження кожного методу з урахуванням особливостей різних типів шумів.

Гаусівський шум характеризується адитивністю та випадковістю, що ускладнює його ефективне усунення без втрати важливих деталей зображення. У цьому дослідженні було проаналізовано три методи фільтрації: медіанний фільтр, вейвлет-фільтр та фільтр Вінера.

- Медіанний фільтр продемонстрував помірне покращення якості обробки зображення зі збільшенням PSNR на +6,52 dB і SSIM на +0,3870. Це вказує на його здатність досить ефективно згладжувати гаусівський шум при помірному спотворенні структури. Однак було відзначено втрати дрібних деталей у текстурованих областях.

- Вейвлет-фільтр досяг найкращих результатів серед протестованих методів: збільшення PSNR склало +7,37 dB, а SSIM - +0,5688. Це свідчить про його високу здатність адаптивно локалізувати шум у частотному просторі та вибірково його пригнічувати. В результаті візуальне сприйняття майже повністю позбавлене шуму, що робить цей метод найбільш ефективним.

- Фільтр Вінера, навпаки показав зниження PSNR на -5,34 dB, хоча SSIM зріс на +0,3222. Це вказує на збереження структурної інформації при значному зменшенні сигналу. Вірогідно, цей метод переоцінив рівень шуму, що спричинило надмірне згладжування зображення і відповідно, погіршення його чіткості.

Імпульсний шум, або ж «сіль і перець», характеризується випадковими пікселями, що мають або максимальні, або мінімальні значення. Через свою дискретність та яскраво виражену природу, цей тип шуму створює різкі контрастні дефекти на зображенні.

- Медіанний фільтр вважається одним із найефективніших способів боротьби з імпульсним шумом, що підтверджується його вражаючими результатами: PSNR збільшився на +18,85 dB, а SSIM зріс на +0,2522. Завдяки локальній обробці даних і стійкості до екстремальних значень, цей метод успішно виглядає шум без пошкодження деталей, таких як контури. Як результат, зображення виглядає чистішим та позбавленим білих і чорних точок.

- Вейвлет-фільтр, на противагу свого успіху з гаусівським шумом, значно поступається в своїй ефективності. Для імпульсного шуму його продуктивність значно знизилась : PSNR впав на 0,89 dB, а SSIM на 0,0521. Основна проблема полягає у неспроможності фільтра точно виявляти та локалізувати шумові пікселі в межах окремих масштабів декомпозиції. Як наслідок, небажані пікселі або залишаються у зображенні, або трансформуються у спотворені артефакти.

- Фільтр Вінера також не забезпечив належної обробки імульсного шуму. PSNR майже не змінився (-0,05 dB), а SSIM навіть погіршився (-0,1851). Це зумовлено припущенням фільтра щодо гаусівської природи шуму, яке не

відповідає властивостям імпульсного шуму. У результаті артефакти або залишаються, або розмиваються, що призводить до втрати чіткості деталей.

Рівномірний шум рівномірно поширюється по амплітудному спектру сигналу, що значно ускладнює його ідентифікацію та усунення. Візуально він менш помітний, проте має вагомий вплив на контрастність і яскравість зображення.

- Медіанний фільтр продемонстрував доволі пристойні результати: значення PSNR збільшилося на 4,13 dB, а SSIM – на 0,3111. Це свідчить про ефективну роботу фільтра з шумом середньої інтенсивності, хоча в областях із високою деталізацією спостерігалося деяке погіршення чіткості.
- Вейвлет-фільтр знову показав найкращу ефективність: приріст PSNR склав 0,91 dB, а SSIM – 0,3774. Завдяки адаптивній обробці шум успішно пригнічується, збереження структури зображення при цьому залишається на високому рівні. Зображення виглядає природно та добре деталізовано.
- Фільтр Вінера продемонстрував змішані результати, подібно до інших типів шуму. Незважаючи на покращення SSIM (+0,1221), значення PSNR знизилося на 10,57 dB. Імовірно, це пов’язано з перенавчанням фільтра, що спричинило занадто агресивне згладжування виглядало тьмяним.

Аналіз дає змогу визначити низку ключових висновків (табл.4.1):

1. Медіанний фільтр демонструє найвищу ефективність у боротьбі з імпульсним шумом, що робить його оптимальним вибором для таких завдань. Він також ефективний для початкового видалення рівномірного та слабкого гаусівського шуму.

2. Вейвлет-фільтр виявився найбільш універсальним у випадках гаусівського та рівномірного шуму. Його здатність проводити аналіз на різних масштабах дозволяє ефективно зберігати суттєву інформацію, одночасно усуваючи перешкоди. Однак його чутливість до типу шуму може суттєво вплинути на результати у разі імпульсного спотворення.

3. Фільтр Вінера потребує точного налаштування для забезпечення ефективності. Неправильна оцінка характеристик шуму може значно знизити

його результативність, тому перед застосуванням необхідна попередня обробка або калібрування.

Таблиця 4.1 – Порівняльна таблиця ефективності фільтрів

Тип шуму	Фільтр	PSNR (після)	SSIM (після)	Поліпшення PSNR	Поліпшення SSIM
Гаусівський	Медіанний	27,31 dB	0,5964	+6,52 dB	+0,3870
	Вейвлет	28,16 dB	0,7775	+7,37 dB	+0,5688
	Вінера	15,44 dB	0,5314	-5,34 dB	+0,3222
Імпульсний	Медіанний	36,33 dB	0,9589	+18,85 dB	+0,2522
	Вейвлет	16,62 dB	0,6554	-0,89 dB	-0,0521
	Вінера	17,43 dB	0,5208	-0,05 dB	-0,1851
Рівномірний	Медіанний	31,69 dB	0,7874	+4,13 dB	+0,3111
	Вейвлет	28,47 dB	0,8537	+0,91 dB	+0,3774
	Вінера	17,00 dB	0,5984	-10,57 dB	+0,1221

Додаткову цікавість становлять комбіновані підходи, наприклад, гібриди вейвлет-медіанних фільтрів або інтеграція класичних методів із алгоритмами машинного навчання. Це дозволить створювати більше гнучких і продуктивних рішень для обробки зображень у реальному часі навіть за складних умов.

Загалом, детальний аналіз процесів фільтрації є фундаментальним для побудови високоякісних систем обробки зображень. Це підтверджується як обчислювальними результатами, так і візуальним спостереженнями.

4.2 Аналіз переваг та недоліків використаних фільтрів.

На основі проведених експериментів з фільтрації ображень, зашумлених різними типами шумів, було виділено три основні фільтри, які демонструють найвищу ефективність: медіанний, фільтр Вінера та вейвлет-фільтр. Кожен із

них має свої переваги, недоліки та область оптимального застосування. Нижче наведено порівняльний аналіз цих методів.

Медіанний фільтр є одним із найпростіших та найефективніших способів приглушення імпульсного шуму («сіль і перець»). Він є нелінійним фільтром, що замінює значення пікселя на медіану в його локальному оточенні.

Переваги:

1. Висока ефективність при обробці зображень, зашумлених імпульсним шумом;
2. Добре зберігає контури, краї та деталі об’єктів;
3. Простота реалізації та низькі обчислювальні витрати.

Недоліки:

1. Неадаптивний: не змінює поведінку в залежності від інтенсивності або типу шуму;
2. Малоефективний для гаусівського або рівномірного шуму;
3. Втрати деталізації при використанні великих вікон.

Рекомендоване застосування:

- Для зашумлених зображень із окремими яскравими пікселями;
- У задачах технічного зору з використаними об’єктами.

Фільтр Вінера є адаптивним підходом до зменшення шумів, який орієнтований на мінімізацію середньоквадратичної похибки між обробленим і вихідним зображеннями. Ця методика базується на оцінці локальних середніх значень і дисперсій, що дозволяє фільтру пристосовуватися до особливостей різних ділянок зображення.

Переваги:

1. Ефективність у боротьбі з гаусівським шумом;
2. Здатність адаптуватися до статистичних властивостей локальних областей;
3. Досягнення балансу між згладжуванням і збереженням структури зображення.

Недоліки:

1. Висока чутливість до точності оцінки параметрів шуму;
2. Низька ефективність при обробці імпульсного шуму;
3. Порівняно вищі обчислювальні витрати у порівнянні з класичними методами фільтрації.

Сфери застосування:

- Обробка зображень із гаусівським або комбінованим шумом;
- Використання в медичній візуалізації, астрономії та наукових дослідженнях.

Вейвлет-фільтр, базується на використанні вейвлет-декомпозиції для розділення сигналу й шуму у спеціальному просторі. Цей метод представляє собою один із найпрогресивніших адаптивних підходів до фільтрації.

Переваги:

1. Висока ефективність при усунені різних типів шуму;
2. Здатність зберігати структуру, текстуру та контури зображення;
3. Можливість багаторівневого аналізу і обробки.

Недоліки:

1. Складність реалізації процесу;
2. Необхідність точного вибору вейвлет-функції та кількості рівнів декомпозиції;
3. Підвищена обчислювальна складність.

Сфера застосування:

- Обробка зображень зі складними текстурами та детальними структурами;
- Високоточна візуалізація, наприклад у географічних системах або біометричних аналізах.

Медіанний фільтр є відмінним рішенням для обробки зображень із наявністю імпульсного шуму. Він поєднує простоту реалізації з високою точністю.

Фільтр Вінера виступає універсальним адаптивним інструментом, що ефективно працює при гаусівському шумі, забезпечуючи стабільну якість результатів.

Вейвлет-фільтрація представляє собою передовий метод, який дозволяє досягти висококласного відновлення навіть за умов складного чи комбінованого шуму. Проте цей підхід вимагає значних обчислювальних ресурсів.

Отже, вибір методу фільтрації повинен залежати від природи шуму, характеристик зображення та можливостей обчислювальної системи. У подальшому тексті будуть також проаналізовані показники швидкодії і перспективи масштабування цих методів.

4.3 Аналіз придатності системи до різних типів зображення.

У ході тестування було проаналізовано ефективність функціонування розробленої системи обробки зашумлених зображень для різних категорій. Перевірка проводилася на наборах цифрових зображень, що охоплюють:

- Побутові фотографії (портрети, пейзажі, предметні кадри) (рис.4.1);
- Медичні знімки (МРТ, рентгенограми у градаціях сірого) (рис. 4.2);
- Технічні ілюстрації (скановані креслення, схеми, текстові документи) (рис. 4.3) ;
- Тестові зображення зі стандартних вибірок (Berkeley, Kodak, USC SIPI) (рис. 4.4).

Рисунок 4.1 – Предметне фото з імпульсним шумом та застосуванням медіанного фільтра.

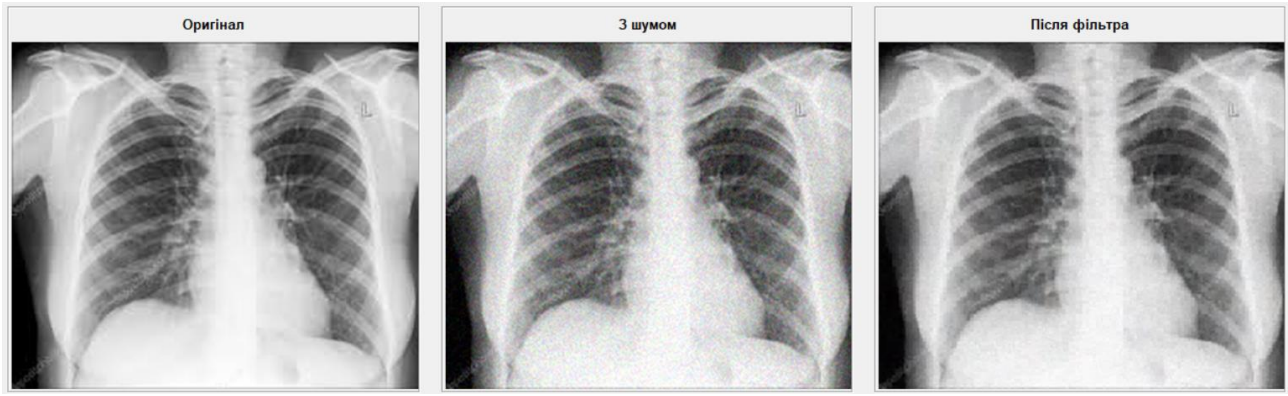

Рисунок 4.2 – Медичне дослідження з рівномірним шумом та застосуванням вейвлет-фільтра.

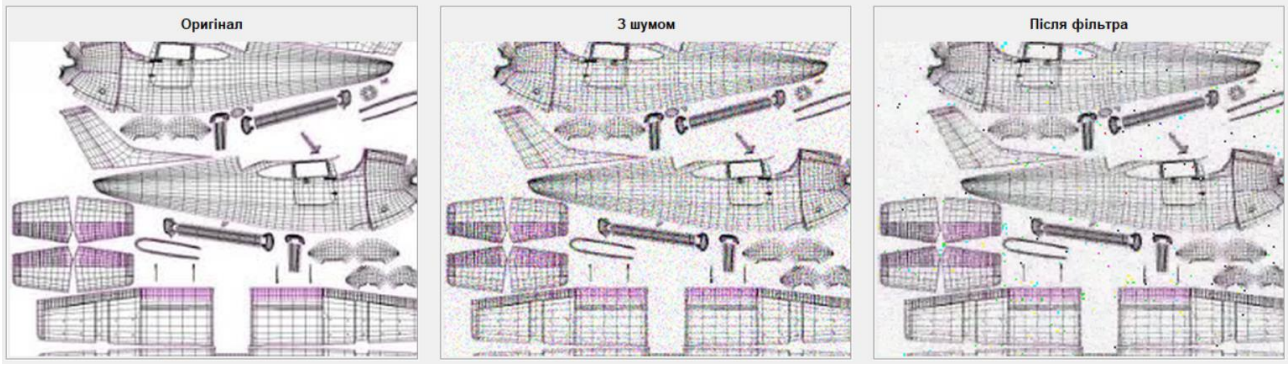

Рисунок 4.3 – Технічна ілюстрація з гаусівським шумом та застосування вейвлет-фільтра.

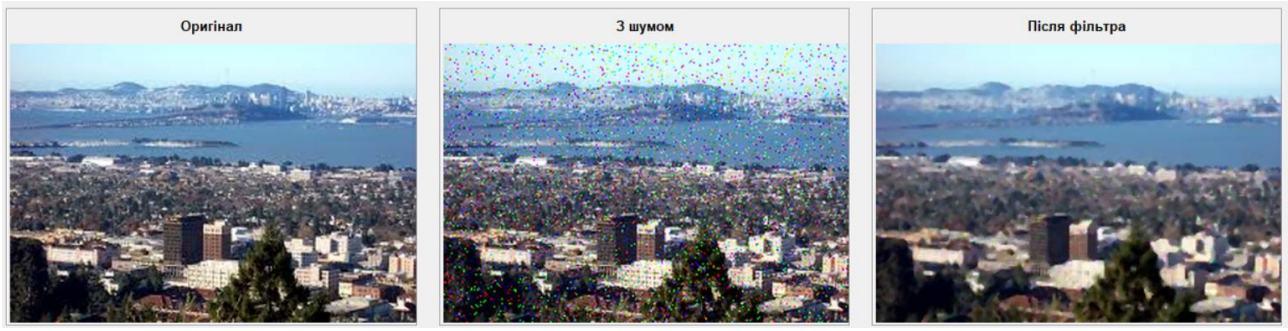

Рисунок 4.4 – Тестове зображення зі стандартних вибірок з імпульсним шумом та застосуванням медіанного фільтра.

Система продемонструвала найвищу продуктивність при роботі із зображеннями, які мають чіткі краї, помірну кількість текстур і деталізовану структуру. Особливо ефективними результати виявилися для:

- Медичних знімків: завдяки адаптивним фільтрам система зберігала діагностично значущі контури, водночас ефективно знижуючи рівень шуму;
- Технічних креслень і документів: медіанний фільтр забезпечив високий рівень обробки імпульсного шуму без втрати дрібних деталей, таких як лінії чи текст;
- Фотографій із рівномірним фоном: поєднання фільтра Вінера та вейвлет-фільтрації дозволило досягти високої якості зображення без суттєвого розмиття.

Втім, система виявила певні обмеження для зображень із високою насиченістю текстур або складним дрібно деталізованим фоном, як-от природні сцени з густим листям чи шерстю тварин. У таких випадках надмірна фільтрація могла призводити до втрати частини структурних елементів, особливо за умов сильного рівня шуму.

У підсумку розроблена система визнана оптимальною для прикладних та інженерних завдань, де пріоритетом є збереження структурної цілісності й контурів об’єктів. Однак її ефективність виявилася обмеженою у випадках роботи з високо текстурованими природними сценами.

4.4 Пропозиції щодо розширення системи.

Розроблена програмна система для обробки зашумлених зображень демонструє високу ефективність у своєму базовому функціоналі, що полягає у фільтрації найбільш поширених типів шуму. Для цього використовуються як класичні, так і адаптивні методи фільтрації, що дозволяють забезпечити задовільний рівень обробки. Однак потенційні можливості вдосконалення цієї системи є вельми широкими, особливо якщо врахувати швидкий розвиток сучасних технологій у таких сферах, як комп'ютерний зір, оптимізація апаратних обчислень та впровадження методів штучного інтелекту. Нижче детально розглянуто ключові напрямки, які можуть стати основою для подальшого розширення функціоналу та підвищення ефективності системи.

Одним із пріоритетних напрямків модернізації є впровадження GPU-оптимізації та підвищення швидкості обчислень. На даному етапі всі алгоритми фільтрації працюють виключно на центральному процесорі (CPU). Це створює значні обмеження щодо продуктивності системи, особливо якщо мова йде про обробку великих за розміром зображень або виконання пакетної обробки. Перехід на використання обчислювальних ресурсів графічного процесора (GPU) може забезпечити суттєве прискорення виконання задач, зокрема адаптивних алгоритмів таких, як Wavelet та Non-Local Means. Для реалізації цієї можливості доцільно інтегрувати відповідні інструменти та бібліотеки, такі як CuPy, PyTorch або OpenCV CUDA, які вже підтримують паралельне виконання обчислень на GPU-архітектурі. У результаті це дозволить системі значно підвищити продуктивність при роботі з великими обсягами даних.

Ще одним важливим напрямком вдосконалення є інтеграція новітніх методів глибокого навчання. Сучасні техніки в області шумозаглушення свідчать про те, що використання глибоких нейронних мереж, таких як DnCNN, FFDNet чи U-Net, дає змогу досягти значно кращої якості відновлення зображень у порівнянні з традиційними підходами. Зокрема, такі моделі демонструють високу ефективність у роботі зі складними та змішаними типами

шумів. Розширення функціоналу на основі впровадження технологій глибокого навчання дозволить не тільки поліпшити якість результату, але й зробити систему більш універсальною та адаптивною до різноманітних сценаріїв використання.

Таким чином, удосконалення розробленої програмної системи має концентруватися на впровадженні сучасних рішень для апаратного прискорення обчислень та інтеграції новітніх алгоритмів машинного навчання. Це дозволить не лише збільшити продуктивність та точність роботи системи, але й забезпечить її відповідність актуальним викликам і потребам галузі комп'ютерного зору.

Для досягнення поставлених цілей можна скористатттатися наступними підходами та технологіями, які сприятимуть ефективному вирішенню завдань обробки зображень:

1. Реалізувати навчання власної нейромережі на основі зашумлених/чистих пар зображень;
2. Інтегрувати попередньо навчені моделі за допомогою бібліотек TensorFlow або PyTorch;
3. Додати інтерфейс для вибору моделі та порівняння з класичними методами.

Щодо подальшого вдосконалення функціоналу, одним із перспективних напрямків є розробка модуля автоматичного визначення типу шуму на зображенні. Наявність такого компонента значно розширить можливості системи, оскільки дозволить їй самостійно обирати найбільш оптимальний фільтр для конкретного зображення. Для побудови такого модуля можна використовувати різні підходи:

1. Застосування статистичного аналізу гістограм для виявлення та ідентифікації імпульсного шуму. Це дозволить на основі розподілу пікселів зробити висновки щодо характеру шуму.

2. Використання спектральних ознак, які дають можливість оцінити частотні компоненти шуму, що забезпечить виявлення специфічних особливостей зображення.

3. Розробка класифікаційних моделей машинного навчання, до яких належать алгоритми Random Forest, метод опорних векторів (SVM) або згорткові нейронні мережі (CNN). Ці моделі дозволяють проводити точну класифікацію різноманітних типів шумів завдяки їх великій здатності до навчання на різнорідних даних.

Одним із ключових напрямків подальшого розвитку можна визначити адаптацію системи для роботи у форматі веб-інтерфейсу або мобільного додатку. Такий підхід не лише значно розширить коло потенційних користувачів, але й зробить систему набагато зручнішою та доступнішою для повсякденного використання, відкриваючи нові горизонти її практичного впровадження.

Реалізація веб-версії системи можлива шляхом використання сучасних технологій веб-розробки. Зокрема, така версія може бути створена із застосуванням Flask або FastAPI, які забезпечують гнучкі й швидкодіючі серверні рішення. Для інтерактивної візуалізації інтерфейсу користувача можуть бути задіяні інструменти типу Gradio чи Streamlit, що відомі своєю простотою інтеграції та потужними можливостями для створення дружнього до користувача середовища.

Що стосується мобільної версії, то її реалізація може базуватися на кількох сучасних інструментах для кросплатформного розроблення. Наприклад, можна використати фреймворк Kivy, який підтримує створення графічних інтерфейсів для мобільних пристроїв. Також можливий вибір на користь React Native, який дозволяє створювати продуктивні й адаптивні додатки для одразу двох провідних мобільних платформ – iOS та Android. Ще одним перспективним варіантом є застосування Flutter – широко популярної технології від Google, яка забезпечує високошвидкісну роботу додатків із нативним виглядом. Усі ці підходи можуть бути ефективно поєднані з Python-

сервером, що виступатиме як база для обробки даних і управління функціональністю, створюючи комплексне та добре інтегроване рішення.

Впроваджена система володіє значним потенціалом для масштабування, як у частині підвищення точності фільтрації, так і в аспекті розширення функціональних можливостей. Запропоновані напрями розвитку сприятимуть формуванню більш адаптивної, ефективної та придатної для практичного застосування системи в професійних галузях, зокрема в медицині, забезпеченні безпеки, використанні дронів і аналізі супутникових даних.

4.5 Перспективи використання в практичних задачах.

Системи обробки зашумлених зображень відкривають надзвичайно великі перспективи для їх використання в різноманітних реальних галузях, де якість візуальної інформації відіграє ключову роль. Висока точність і ефективність таких систем стають вирішальними для досягнення поставлених завдань у ситуаціях, що потребують детального аналізу візуальних даних. Запропоноване рішення, яке базується на інтеграції класичних методів фільтрації та адаптивних підходів до очищення зображень, здатне забезпечувати максимальну якість результатів. Ця технологія може функціонувати як автономний інструмент або ж стати важливим елементом ширших комплексних інтелектуальних систем. Далі представлені ключові напрями практичного застосування розробленої системи, які чітко демонструють її значимість у сучасних технологічних процесах.

У сфері охорони здоров’я якість зображень, отриманих за допомогою медичних візуалізаційних пристроїв, таких як магнітно-резонансна томографія (МРТ), комп’ютерна томографія (КТ) чи ультразвукове дослідження (УЗД), безпосередньо впливає на точність діагностики, а також на своєчасність і правильність призначеного лікування. Часто ці зображення супроводжуються шумами через вплив електронних перешкод, недосконалості сенсорного

обладнання або навіть рух пацієнта під час процедури. Використання запропонованої системи може значно підвищити якість обробки таких даних завдяки трьом ключовим підходам:

- Проведення попереднього очищення зображень, отриманих при МРТ і КТ;
- Покращення чіткості УЗД-зображень для більш точної оцінки структур внутрішніх органів;
- Мінімізація необхідності повторного проведення діагностичних процедур, що часто спричиняється низькою якістю первинного знімка.

Система також знаходить своє застосування у сфері дистанційного зондування та геоінформаційних технологій. Зображення, отримані за допомогою супутників або безпілотних авіаційних апаратів (дронів), нерідко страждають від впливу різних зовнішніх змінних, таких як атмосферні перешкоди, неоднорідність освітлення чи радіосигнальні завади.

Крім того, при обробці складніших гіперспектральних зображень у галузі дистанційного зондування адаптивна фільтрація виявляється дуже корисною, оскільки вона дозволяє ефективно виділити потрібну інформацію у багатовимірних спектральних даних. Це значно полегшує проведення детального аналізу територій та оптимізує подальші дослідницькі чи виробничі процеси.

Шум у відеозаписах, отриманих з камер спостереження, може виникати через кілька факторів, таких як недостатнє освітлення, погодні умови або технічні обмеження обладнання. У подібних випадках система надає можливості:

- Покращення якості кадрів для чіткої ідентифікації осіб;
- Забезпечення якісного запису в умовах слабкого освітлення чи вночі;
- Підвищення точності роботи алгоритмів розпізнавання руху чи об’єктів.

У процесах реставрації та оцифрування старих зображень, наприклад, друкованих матеріалів, фотографій або архівних документів, які часто містять чимало шумів після сканування, система може бути застосована для:

- Автоматичного очищення оцифрованих документів;
- Усунення шумів відносно плівки чи друку;
- Покращення зображень для їх використання у цифрових архівах.

Розроблена система фільтрації шуму демонструє універсальність у застосуванні до різних практичних завдань. Її можна адаптувати до конкретної проблематики шляхом налаштування параметрів фільтрів, додавання нових алгоритмів або інтеграції в масштабні інформаційні системи. Окрім цього, подальший розвиток системи в напрямках GPU-оптимізації, впровадження методів глибокого навчання та багатоплатформної сумісності відкриває шлях до її використання як у промислових процесах, так і у сфері наукових досліджень.

ВИСНОВКИ

У рамках виконання бакалаврської кваліфікаційної роботи було створено програмну систему для обробки зашумлених зображень із використанням адаптивних фільтрів, реалізовану мовою програмування Python. Головна мета полягала в підвищенні якості цифрових зображень шляхом ефективного видалення різноманітних видів шумів, зберігаючи при цьому важливі деталі візуальної інформації.

Під час аналізу предметної області було ретельно досліджено найбільш поширені типи шумів, зокрема гаусівський, імпульсний та рівномірний. З'ясовано, що кожен із цих типів має специфічні особливості та по різному впливає на якість зображень, що вимагає впровадження спеціалізованих методів фільтрації для досягнення найкращого результату.

З огляду на проведені дослідження, було обґрунтовано доцільність застосування адаптивних методів фільтрації, таких як медіанний фільтр, фільтр Вінера та вейвлет-фільтр. Кожен із зазначених методів інтегровано в систему, враховуючи характеристики певного виду шуму, що дало змогу забезпечити гнучкий і точний підхід до обробки зображень.

Розроблена система базується на модульній архітектурі, яка охоплює ключові компоненти: інтерфейс користувача, модулі завантаження зображень, додавання шуму, фільтрацію, візуалізацію результатів і розрахунок метрик PSNR та SSIM для оцінки якості обробки. Такий підхід забезпечує зручність експлуатації та створює основу для подальшого розширення функціональних можливостей.

Під час тестування на різних типах зображень було підтверджено ефективність застосованих методів фільтрації. Результати свідчать про доцільність використання адаптивних методів для обробки зображень у реальних умовах, де шуми мають змішану або неоднорідну природу. Система демонструє високу гнучкість і здатність адаптуватися до змін витакмог, що

робить її придатною для інтеграції в ширший спектр програмних рішень у сфері комп’ютерного зору.

У цілому розроблена платформа може стати основою для створення складніших систем обробки зображень. ЇЇ потенціал є особливо цінним у таких галузях, як медична діагностика, відеоспостереження або автономні технології, де критично важлива висока якість візуальної інформації.

Відкритість архітектури та використання мови Python забезпечують легкість інтеграції нових фільтрів, алгоритмів попередньої обробки або додаткових метрик оцінювання, що створює широкі можливості для подальших досліджень і вдосконалення. Система може бути налаштована для роботи з потоковими даними або підключення до зовнішніх джерел, таких як камери спостереження чи медичне обладнання. Завдяки своїй гнучкості вона однаково добре підходить як для навчальних цілей у вивченні методів обробки зображень, так і для професійного використання в точних сферах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аналіз поширених методів накладання шуму на зображення. Н.Б. Шаховська, О.І. Косар. Національний університет "Львівська політехніка", м. Львів, Україна. URL: https://nv.nltu.edu.ua/Archive/2018/28_1/31.pdf

2. Порівняння алгоритмів фільтрації медичних зображень за оцінками їх якості / Бондіна Н.М., Калмичков О.С., Козіна О.А. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2013. – No 39 (1012). – С. 15 – 21. URL: https://science.lpnu.ua/uk/scsit/vsi-vypusky/vypusk-864-nomer-1-2017/adaptyvnyy-metod-poperednoyi-obrobky-gistologichnyh-ta

3. Адаптивний метод попередньої обробки гістологічних та цитологічних зображень. Вісник НУ «Львівська політехніка» Комп'ютерні науки та інформаційні технології, № 864 URL: https://www.researchgate.net/publication/322028552_ADAPTIVNIJ_METOD_POPEREDNOI_OBROBKI_GISTOLOGICNIH_TA_CITOLOGICNIH_ZOBRAZEN

4. Yinpeng J. Contrast Enhancement by Multi-scale Adaptive Histogram Equalization / Yinpeng Jina, Laura Fayadb, Andrew Laine // Proceedings of SPIE Vol. 4478 (2001), pp 206-213 URL: https://academiccommons.columbia.edu/doi/10.7916/D8QZ2M29

5. Гуліч, Є. Д. Фільтрація напівтонових зображень спотворених адитивним білим гаусівським шумом : дипломна робота ... бакалавра : 172 Радіотехнічні інформаційні технології / Гуліч Єгор Дмитрович. – Київ, 2020. – 80 с. URL: https://ela.kpi.ua/items/bb9e6440-ad12-4f99-a317-11c0460d26ca

6. Аналіз поширених методів накладання шуму на зображення. / Шаховська. Н. Б., Косар. О. І.,2018. – 146 с.

7. Сучасні уявлення про імпульсний шум інженерно-техічного обладнання. О.І.Сороковий, к.т.н., доц., Полтавський національний технічний університет імені Юрія Кондратюка. URL: https://reposit.nupp.edu.ua/bitstream/PoltNTU/8224/1/Znpgmb_2010_2_39.pdf

8. Цифрова обробка зображень [Текст] : метод, рекомендації до викон. лаборатор. робіт для студ. спеціальності 7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка» усіх форм навчання / Уклад.: В. С. Лазебний, П. В. Попович. - К.: НТУУ «КПІ», 2016. - 73 с URL: https://ela.kpi.ua/server/api/core/bitstreams/2301c9b9-f626-4c13-801c-5b700f77a8f3/content

9. Бодян І.С. Софина О.Ю. Реалізація комплексної системи обробки зашумлених зображень з використанням адаптивних фільтрів на Python. – ВНТКП ВНТУ. Факультет інтелектуальних інформаційних технологій та автоматизації LIV Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2025). URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23119/19181

10. Половинко Ігор, Семочко Олександр. Метод оцінки відновлення зображень із використанням просторових і частотних фільтрів. International Science Journal of Engineering & Agriculture. Vol. 1, No. 4, 2022, pp. 8-18. doi: 10.46299/j.isjea.20220104.02 URL: https://isg-journal.com/isjea/article/download/83/68

11. Що таке шум під час обробки зображень? – Буквар. Електронний ресурс. URL: https://www.unite.ai/uk/what-is-noise-in-image-processing-a-primer/

12. Інтелектуалізована система оцінювання динамічних змін біомедичних зображень :монографія / Н.П. Бабюк, С.В. Павлов, П.Ф. Колісник, О.Ж. Мамирбаєв, Ж.Ж. Ажибєкова. – Вінниця : ВНТУ, 2024. – 131 с. URL: https://press.vntu.edu.ua/index.php/vntu/catalog/book/846

13. Вейвлет технології у застосуванні до обробки теплових зображень. Електронний ресурс. URL: https://iroptika.at.ua/news/vejvlet_tekhnologiji/2014-07-28-17

14. Комп’ютерне моделювання систем та процесів. Методи обчислення. Частина 2. Квєтний Р.Н., Богач І.В., Бойко О.Р., Соіина О.Ю., Шушура О.М. Електронний ресурс. URL:

https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t2/213..htm

15. Лавер В.О., Левчук О.М. Обробка зображень: навч.-метод. посіб. / В.О. Лавер, О.М. Левчук. – Ужгород : вид-во ПП «АУТДОР - ШАРК», 2021. – 51 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/35667/1/%D0%9E%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C%202021.pdf

16. Вступ до OpenCV. Комп’ютерний зір. Інформаційний ресурс про електроніку та програмування. Електронний ресурс. URL: https://itmaster.biz.ua/programming/vision/opencv.html

17. OpenCV – визначення контурів на зображенні. Інформаційний ресурс про електроніку та програмування. Електронний ресурс. URL: https://itmaster.biz.ua/programming/vision/opencv-edges.html

18. Bradski G., Kaehler A., «Learning OpenCV: Computer vision with the OpenCV library», O'Reilly Media, Inc., 2008

19. Culjak I., "A brief introduction to OpenCV." 2012 proceedings of the 35th international convention MIPRO. IEEE, 2012.

20. Огляд пакету Python NumPy. LibreTexts Ukrayinska. Електронний ресурс. URL: https://surl.li/zwnned

21. Чому вам слід використовувати NumPy. Dev Zone. Електронний ресурс. URL: https://devzone.org.ua/post/chomu-vam-slid-vykorystovuvaty-numpy

22. Бібліотека Pillow: обробка зображень у Python. Highload. Електронний ресурс. URL: https://highload.tech/uk/biblioteka-pillow-obrobka-zobrazhen-u-python/

23. ВИКОРИСТАННЯ ВІРТУАЛЬНИХ МЕТРИК ДЛЯ ОЦІНКИ ЯКОСТІ КОЛЬОРОВИХ ЗОБРАЖЕНЬ В. П. Ткаченко, А. С. Гордєєв Харківський національний університет радіоелектроніки, просп. Науки, 14, Харків, 61166, Україна.

24. Sunyoung, Cho, & Hyeran, Byun. (2022). Dynamic curve color model for image matting: Pattern Recognition Letters, 33, 920–933 (in English)

25. ВИКОРИСТАННЯ ВІЗУАЛЬНИХ МЕТРИК ДЛЯ АНАЛІЗУ СТИСНЕННЯ З ВТРАТАМИ ЗАШУМЛНИХ ЗОБРАЖЕНЬ Б. В. Коваленко, В.В. Лукін Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Україна. URL: https://www.researchgate.net/publication/357286876_Vikoristanna_vizualnih_metrik_dla_analizu_stisnenna_z_vtratami_zasumlenih_zobrazen

ДОДАТКИ

Додаток А

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА КОМПЛЕКСНОЇ СИСТЕМИ ОБРОБКИ ЗАШУМЛЕНИХ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ АДАПТИВНИХ ФІЛЬТРІВ PYTHON

Алгоритм роботи системи

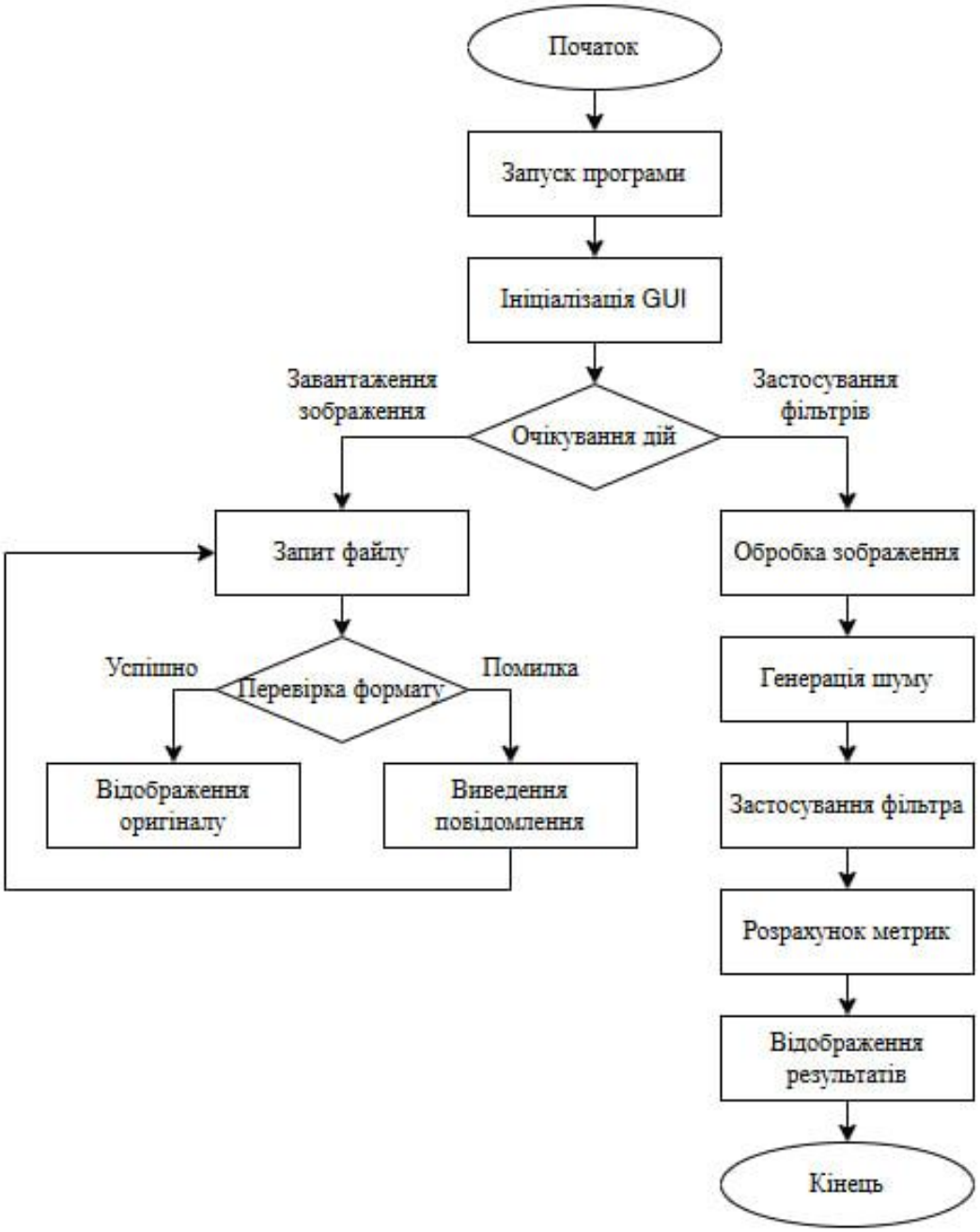

Рисунок А.1 – Алгоритм роботи системи.

Схема інтеграції із зовнішніми системами

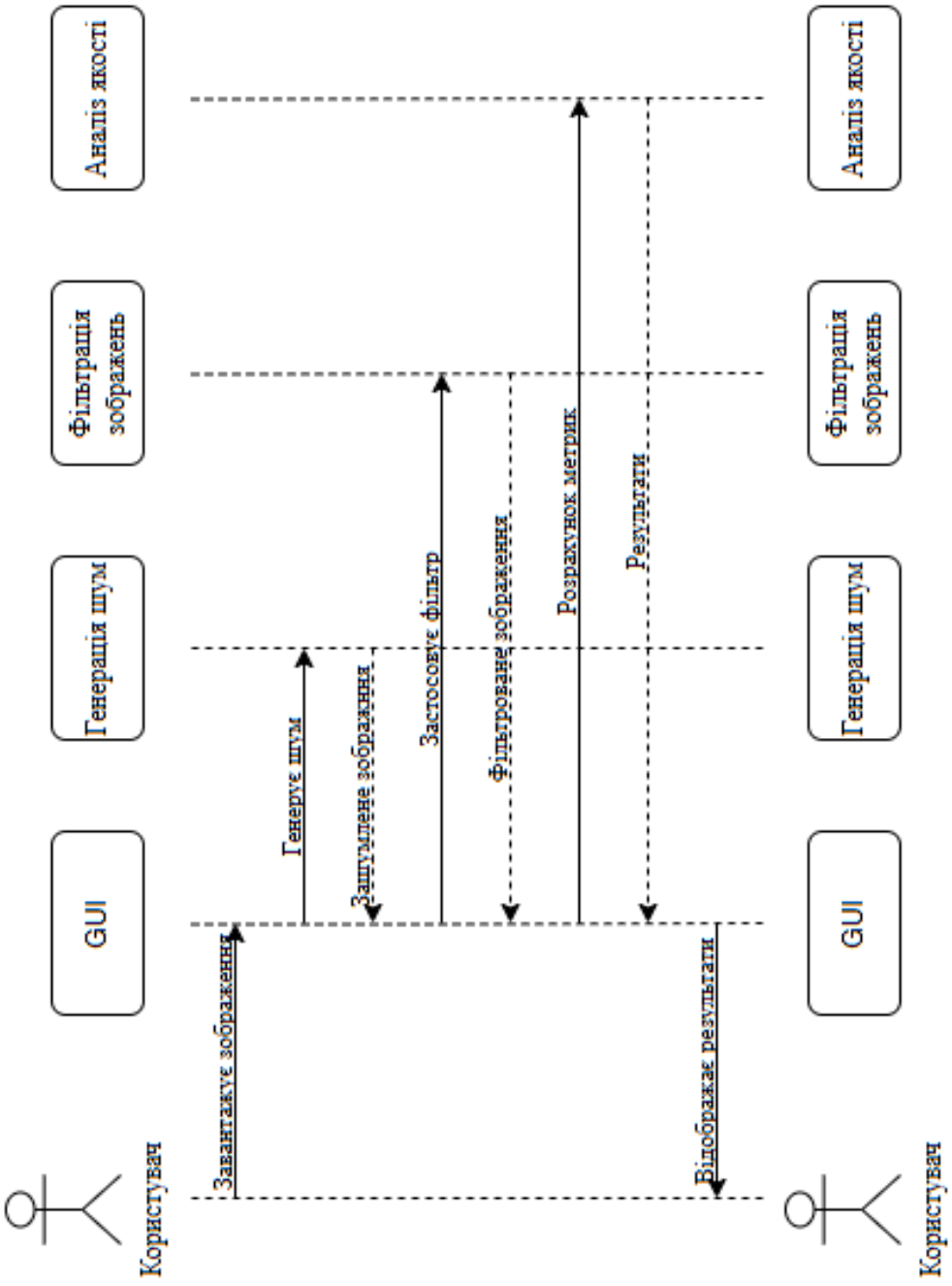

Рисунок А.2 – Схема інтеграції з зовнішніми системами.

Діаграма компонентів

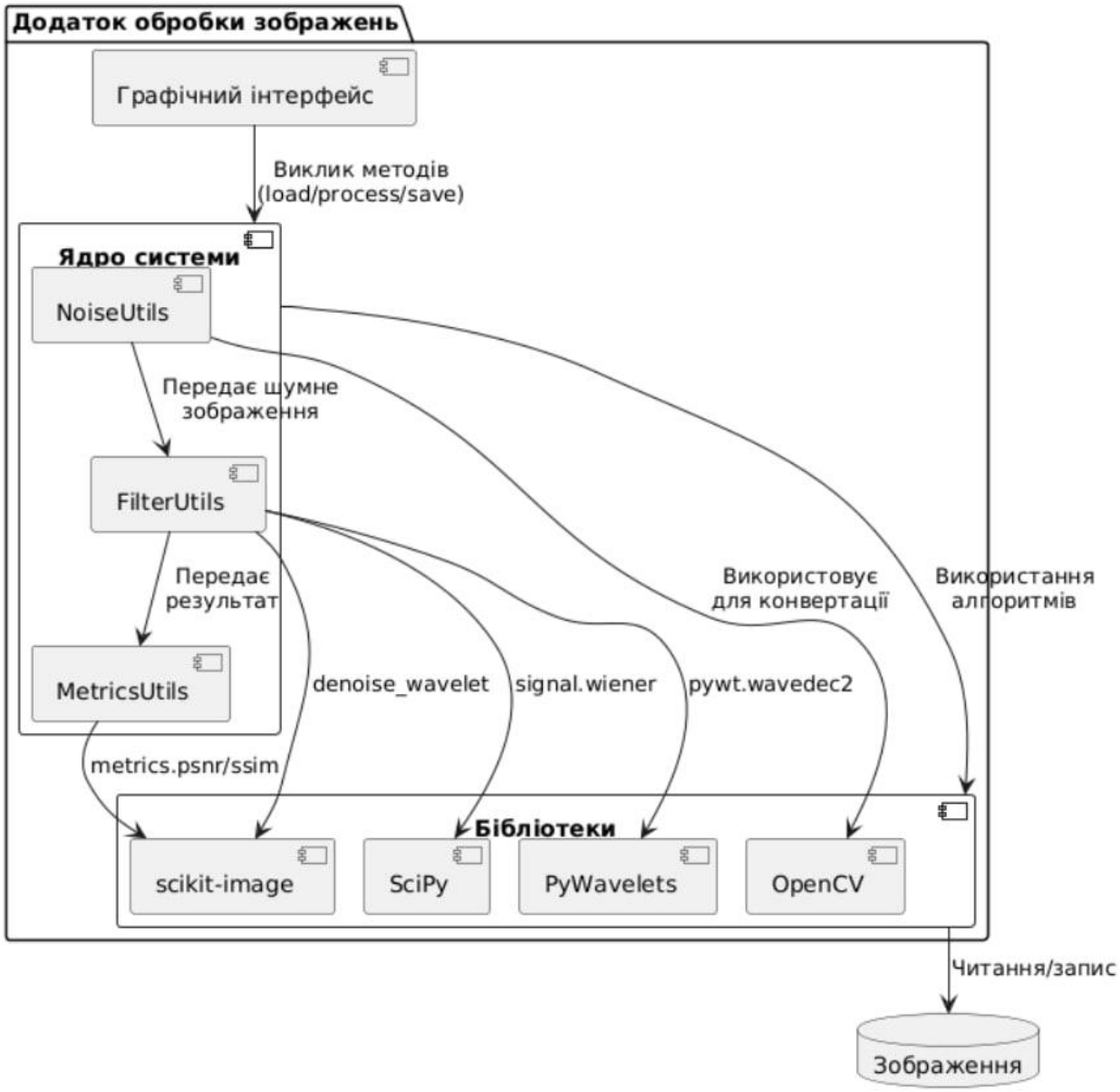

Рисунок А.3 – Діаграма компонентів.

Додаток Б

(обов’язковий)

Лістинг основних функцій програми

```
import numpy as np
import cv2
from skimage.util import random_noise

def load_image(path):
    img = cv2.imread(path)
    return cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

def add_gaussian_noise(image, mean=0, var=0.01):
    noisy = random_noise(image, mode='gaussian', mean=mean, var=var)
    return (255 * noisy).astype(np.uint8)

def add_salt_pepper_noise(image, amount=0.05, salt_vs_pepper=0.5):
    noisy = random_noise(image, mode='s&p', amount=amount, salt_vs_pepper=salt_vs_pepper)
    return (255 * noisy).astype(np.uint8)

def add_uniform_noise(image, intensity=20):
    noise = np.random.uniform(-intensity, intensity, image.shape).astype(np.int16)
    noisy = image.astype(np.int16) + noise
    noisy = np.clip(noisy, 0, 255)
    return noisy.astype(np.uint8)

import cv2
import numpy as np
```

```
import scipy.signal
import pywt
from skimage.restoration import denoise_wavelet, wiener

def apply_median_filter(image, kernel_size=3):
    return cv2.medianBlur(image, kernel_size)

def apply_wavelet_filter(image):
    image = image.astype(np.float32) / 255.0

    if image.ndim == 3:
        denoised = denoise_wavelet(
            image,
            channel_axis=-1,
            convert2ycbcr=True,
            rescale_sigma=True
        )
    else:
        denoised = denoise_wavelet(
            image,
            channel_axis=None,
            rescale_sigma=True
        )

    return (denoised * 255).astype(np.uint8)

def apply_wiener_filter(image, kernel_size=5):
    if image.ndim == 3:
```

```
        return np.stack([
            scipy.signal.wiener(image[:, :, i], (kernel_size, kernel_size))
            for i in range(3)
        ], axis=2).astype(np.uint8)
    else:
        return scipy.signal.wiener(image, (kernel_size, kernel_size)).astype(np.uint8)

from skimage.metrics import peak_signal_noise_ratio, structural_similarity
import numpy as np

def compute_psnr(original, processed):
    return peak_signal_noise_ratio(original, processed)

def compute_ssim(original, processed):
    if original.ndim == 3:
        ssim_val = structural_similarity(original, processed, channel_axis=-1, win_size=3)
    else:
        ssim_val = structural_similarity(original, processed, win_size=3)
    return ssim_val

def evaluate_filters(original, noisy, filters_dict):
    results = []

    for name, filtered in filters_dict.items():
        psnr_val = compute_psnr(original, filtered)
        ssim_val = compute_ssim(original, filtered)
        results.append({
            'Фільтр': name,
```

```
        'PSNR': round(psnr_val, 2),
        'SSIM': round(ssim_val, 4)
    })

return results
```

87

Додаток В
(обов'язковий)
ПРОТОКОЛ
ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Реалізація комплексної системи обробки зашумлених зображень з використанням адаптивних фільтрів на Python

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра АІІТ, ФІТА, 1ІСТ-21Б
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1,74%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☑ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:
Олег БЕЛЕЛО, зав. каф. АІІТ
(прізвище, ініціали, посада) (підпис)
Любов ЯНЧУК, секретар каф. АІІТ
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку (підпис) Овчинников К.В. (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник (підпис) Софина О.Ю. (прізвище, ініціали, посада)

Здобувач (підпис) Бодян І.С. (прізвище, ініціали)