

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та
автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматизації та інтелектуальних інформаційних технологій

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка модуля фільтрації цифрових зображень для системи
автоматизації відділу відеоаналітики підприємства»

Виконав: студент 1-го курсу, групи
1АКІТР-23м спеціальності 174 –
Автоматизація, комп'ютерно-
інтегровані технології та роботехніка

(шифр і назва, напряму підготовки, спеціальності)

Андрій САЛІМОНОВИЧ

(ім'я та прізвище)

Керівник: к.т.н., доцент каф. АІТ

Володимир ГАРМАШ

(ім'я та прізвище)

«15» 12 2024 р.

Рецензент: д.т.н., доцент каф. КСУ

Марія ЮХИМЧУК

(ім'я та прізвище)

«15» 12 2024 р.

Допущено до захисту

Зав. кафедри АІТ, д.т.н., проф.

Олег БІСІКАЛО

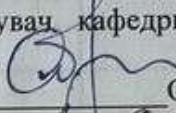
(ім'я та прізвище)

«16» 12 2024 р.

Вінниця ВНТУ - 2024 рік

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти другий (магістерський)
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність – 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
Освітньо - професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри АПТ, д.т.н.,
проф. 

Олег БІСІКАЛО

"19" березня 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
студенту Салімоновичу Андрію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка модуля фільтрації цифрових зображень для
системи автоматизації відділу відеоаналітики підприємства
керівник роботи Гармаш Володимир Володимирович, доцент кафедри
АПТ

затверджені наказом ВНТУ №310 від 17.09.2024 року

2. Термін подання студентом роботи 16.12. 2024 року

3. Вихідні дані до роботи:

тип цифрового зображення – півтонові та кольорові; мінімальна
роздільна здатність зображення 128 x 128 пікселів, максимальна роздільна
здатність 2048 x 2048 пікселів; рівень шуму – 5 – 70%.

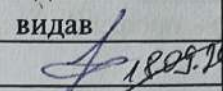
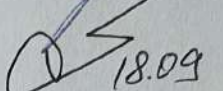
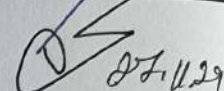
4. Зміст текстової частини: вступ, аналіз існуючих модулів та методів
фільтрації цифрових зображень, розробка методу фільтрації на основі
білатерального фільтру, розробка алгоритмічного та програмного
забезпечення, економічна частина, висновки, список використаних джерел,
додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням
обов'язкових креслень)

Процес фільтрації кратномасштабним білатеральним фільтром. Метод
фільтрації на основі білатерального фільтру. Схема програми. Процес
фільтрації цифрових зображень зі штучним шумом. Схема програми. Процес
фільтрації цифрових зображень з просторово-випадковим шумом. Схема
програми. Структура роботи підсистеми відеоспостереження з модулем
фільтрації зображень. Діаграма класів процесів фільтрації цифрових

зображень зі штучним шумом. Діаграма класів методики дослідження модуля фільтрації цифрових зображень. Use case діаграма процесу фільтрації цифрового зображення.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Гармаш В.В. доц. каф. АІТ	 18.09.24	 18.09.24
4	Козловський В.О. професор кафедри ЕПВМ	 18.09	 27.11.24

7. Дата видачі завдання 18.09.2024 року

КАЛЕНДАРНИЙ ПЛАН

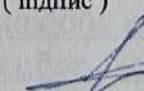
№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз існуючих модулів та методів фільтрації цифрових зображень	до 11.10.2024 р.	<i>вик</i>
2	Розробка методу фільтрації на основі білатерального фільтру	до 26.10.2024 р.	<i>вик.</i>
3	Розробка алгоритмічного та програмного забезпечення	до 15.11.2024 р.	<i>вик</i>
4	Економічна частина	до 27.11.2024 р.	<i>вик</i>
5	Оформлення пояснювальної записки	до 01.12.2024 р.	<i>вик</i>
6	Попередній захист роботи	до 05.12.2024 р.	<i>вик</i>
7	Остаточний захист роботи	до 20.12.2024 р.	<i>вик</i>

Студент


(підпис)

Андрій САЛІМОНОВИЧ
(ім'я і ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Володимир ГАРМАШ
(ім'я і ПРІЗВИЩЕ)

АНОТАЦІЯ

УДК 004.932

Салімонович А. О. Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства. Магістерська кваліфікаційна робота зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2024. 134 с.

На укр. мові. Бібліогр.: 41 назва; рис.: 31; табл. 17.

У магістерській кваліфікаційній роботі розроблено модуль фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства. В оглядово-аналітичній частині розглянуто сучасні методи фільтрації цифрових зображень, зокрема особливості їхнього застосування в автоматизованих системах відеоаналітики, та обґрунтовано доцільність використання методу на основі білатерального фільтру. У теоретично-методичній частині запропоновано новий метод фільтрації, який враховує якісні характеристики зображень, та розроблено відповідний програмний модуль. У практичній частині описано вибір інструментальних засобів реалізації, розроблено та протестовано програмне забезпечення для модуля фільтрації. Економічна частина містить оцінку доцільності впровадження, аналіз технічного рівня, а також розрахунок витрат на розробку та інтеграцію модуля у систему автоматизації. Ілюстративна частина представлена матеріалами, що відображають основні етапи реалізації розробки.

Ключові слова: автоматизована система, обробка зображень, відеоспостереження, відділ відеоаналітики, модуль фільтрації, програма.

ABSTRACT

Salimonovich A. O. Development of a digital image filtering module for the automation system of the video analytics department of the enterprise. Master's qualification work in the specialty 174 "Automation, computer-integrated technologies and robotics", educational program - Intelligent computer systems. Vinnytsia: VNTU, 2024. 134 p.

In Ukrainian. Bibliography: 41 titles; fig.: 31; table. 17.

In the master's qualification work, a digital image filtering module is developed for the automation system of the video analytics department of the enterprise. The review and analytical part considers modern methods of filtering digital images, in particular, the features of their application in automated video analytics systems, and justifies the feasibility of using a method based on a bilateral filter. In the theoretical and methodological part, a new filtering method is proposed that takes into account the qualitative characteristics of images, and a corresponding software module is developed. In the practical part, the choice of implementation tools is described, software for the filtering module is developed and tested. The economic part contains an assessment of the feasibility of implementation, an analysis of the technical level, as well as a calculation of the costs of developing and integrating the module into the automation system. The illustrative part is presented with materials that reflect the main stages of development implementation.

Keywords: automated system, image processing, video surveillance, video analytics department, filtering module, program.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ІСНУЮЧИХ МОДУЛІВ ТА МЕТОДІВ ФІЛЬТРАЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ	7
1.1 Загальна характеристика задачі	7
1.2 Класифікація методів покращення зображень	9
1.3 Аналіз методів фільтрації цифрових зображень	15
1.3.1 Локальні методи фільтрації	17
1.3.2 Методи, що ґрунтуються на диференціальних рівняннях з частковими похідними	21
1.3.3 Фільтри в частотній області	24
1.4 Висновки до першого розділу	27
2 РОЗРОБКА МЕТОДУ ФІЛЬТРАЦІЇ НА ОСНОВІ БІЛАТЕРАЛЬНОГО ФІЛЬТРУ	29
2.1 Дискретне вейвлет-перетворення.....	30
2.2 Опис методу обробки цифрових зображень за допомогою білатерального фільтра	33
2.3 Експериментальні дослідження розробленого методу.....	40
2.4 Розроблення узагальненої методики дослідження модуля фільтрації цифрових зображень	55
2.5 Висновки до другого розділу	60
3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	62
3.1 Розробка алгоритмічного забезпечення.....	62
3.2 Розробка програмного забезпечення.....	70
3.3 Висновки до третього розділу	72
4. ЕКОНОМІЧНА ЧАСТИНА	74
4.1 Проведення комерційного та технологічного аудиту науково- технічної розробки	75

4.2 Розрахунок узагальненого коефіцієнта якості розробки.....	78
4.3 Розрахунок витрат на проведення науково-дослідної роботи	81
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором..	98
4.5 Висновки до розділу	104
ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	107
ДОДАТКИ	112
Додаток А (обов'язковий). Технічне завдання	113
Додаток Б (обов'язковий). Графічна частина	117
Додаток В (обов'язковий). Лістинг модуля програми	125
Додаток Г (обов'язковий). Протокол перевірки дипломної роботи...	134

ВСТУП

Сьогодні цифрова обробка зображень (ЦОЗ) займає важливе місце у багатьох сферах, таких як телекомунікації, системи відеоспостереження, медичні дослідження, робототехніка та автоматизація. Постійне зростання обсягів даних, необхідність у покращенні якості зображень і робота за умов низького співвідношення сигналу до шуму спонукають до створення нових алгоритмів і методів у галузі ЦОЗ.

Особливої актуальності набуває задача покращення якості зображень у системах автоматизації, які використовують відеоаналітику. У таких системах, особливо за умов слабкого освітлення або наявності значного шуму, якість зображень може суттєво знижуватися. Це впливає на точність аналізу, класифікації та прийняття рішень, знижуючи ефективність системи в цілому. Шум, що виникає в процесі отримання та оброблення зображень, також ускладнює стискання даних і зменшує частку корисної інформації.

Обробка зображень за допомогою фільтрації є ключовим етапом підготовки даних до подальшого аналізу. Різноманіття методів фільтрації дозволяє врахувати їхні сильні та слабкі сторони, оскільки вони зазвичай спрямовані на усунення конкретних типів шуму. Однак універсального рішення для всіх видів шуму не існує, тому вибір підходу визначається залежно від поставленої задачі.

Таким чином, робота, присвячена розробці модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства, є актуальною.

Об'єктом дослідження є процес фільтрації цифрових зображень у системах відеоаналітики.

Предметом дослідження є методи і програмні рішення для фільтрації зображень.

Метою роботи є підвищення ефективності обробки даних у системах автоматизації відділу відеоаналітики підприємства шляхом розробки модуля фільтрації цифрових зображень.

Для досягнення мети поставлено такі завдання:

- провести аналіз сучасних методів фільтрації цифрових зображень;
- розробити метод фільтрації, адаптований для потреб відеоаналітики;
- реалізувати алгоритмічне та програмне забезпечення модуля фільтрації;
- провести тестування і оцінку ефективності розробленого модуля.

Науково-технічний результат

Розроблено метод фільтрації шумів у півтонових зображеннях, який поєднує білатеральний фільтр із вейвлет-банком фільтрів, що забезпечило покращення співвідношення сигнал/шум у середньому на 0,1–1 дБ.

Практичне значення одержаних результатів

Практична значущість отриманих результатів полягає у розробці та впровадженні програмного модуля для фільтрації цифрових зображень у системах автоматизації відеоаналітики на підприємстві. Запропонований модуль фільтрації дозволяє покращити якість зображень, особливо тих, що були отримані за умов слабого освітлення або високого рівня шуму. Завдяки спеціально адаптованим алгоритмам, створений модуль ефективно знижує рівень шуму на цифрових зображеннях, забезпечуючи точність для потреб відеоаналітики. Запропоновані розробки можуть бути впроваджені в системах автоматизації для забезпечення високої якості обробки вхідних даних, що є критично важливим для автоматизованого відеоаналізу, створення довготривалих архівів, а також підвищення загальної ефективності роботи відділу відеоаналітики.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції ВНТУ: «Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)».

Публікації. Салімонович Андрій Олександрович, Гармаш Володимир Володимирович «Метод фільтрації цифрових зображень на основі білатерального фільтру», «Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)», 2024.

URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20377>

1 АНАЛІЗ ІСНУЮЧИХ МОДУЛІВ ТА МЕТОДІВ ФІЛЬТРАЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1 Опис об'єкта автоматизації та загальна характеристика задачі

Відділ відеоаналітики підприємства є ключовим підрозділом, що займається обробленням, аналізом і зберіганням відеоданих, отриманих із систем відеоспостереження, виробничих процесів або інших джерел. Основними задачами відділу є:

- виявлення і відстеження подій у реальному часі;
- класифікація та інтерпретація відеоданих;
- створення аналітичних звітів на основі отриманих результатів;
- забезпечення високої якості відеоархівів для довгострокового зберігання та подальшого використання.

У роботі відділу важливу роль відіграє якість відеозображень. Проблеми, пов'язані з низьким рівнем освітленості, наявністю шумів або артефактів у відеоматеріалах, негативно впливають на точність і швидкість виконання задач відеоаналітики. Тому виникає потреба у впровадженні спеціалізованих методів попередньої обробки відеоданих, зокрема фільтрації зображень, для забезпечення їхньої якості.

Модуль фільтрації цифрових зображень є необхідним для підвищення ефективності роботи відділу відеоаналітики. Його розробка зумовлена такими причинами:

1. Покращення якості вхідних даних: Фільтрація дозволяє усунути шум і артефакти, що виникають під час запису чи передачі відео. Це сприяє підвищенню точності алгоритмів розпізнавання і аналізу.
2. Оптимізація роботи аналітичних систем: Зображення високої якості спрощують виконання задач автоматичного розпізнавання об'єктів, аналізу руху та інших процесів відеоаналітики.

3. Ефективне управління відеоархівами: Завдяки фільтрації створюється можливість зберігати більш якісні відеодані, що полегшує їх використання у майбутньому.
4. Підвищення продуктивності відділу: Автоматизація процесу попередньої обробки зображень знижує трудовитрати співробітників, дозволяючи зосередитися на аналізі даних і прийнятті рішень.

Розробка модуля фільтрації дозволить забезпечити високий рівень попередньої обробки зображень, що стане основою для підвищення ефективності роботи відділу відеоаналітики та покращення якості виконуваних задач.

Методи фільтрації даних мають ключове значення у завданнях передачі, обробки та зберігання відеозображень. Це зумовлено тим, що цифрові потоки в таких системах зазвичай мають високу інтенсивність, а якість обробки відеоданих істотно впливає на продуктивність і ефективність роботи автоматизованих систем. Значна частина оброблюваної інформації припадає на відеозображення та послідовності кадрів, які часто супроводжуються шумами, артефактами або викривленнями, що виникають через несприятливі умови зйомки, недоліки технічного обладнання чи вплив завад.

Ефективне фільтрування даних є важливим етапом для зменшення шуму та забезпечення високої візуальної якості відеозображень. Це критично важливо, оскільки якість оброблених даних безпосередньо впливає на виконання задач автоматизованого відеоаналізу, таких як пошук і ідентифікація об'єктів, аналіз подій або оцінка кількісних характеристик. Зниження рівня шуму сприяє підвищенню точності роботи алгоритмів обробки відео, покращенню ефективності використання ресурсів і забезпеченню високої якості збереження архівів. Таким чином, проблема видалення шуму та покращення якості відеозображень є однією з актуальних задач у сфері обробки цифрових даних, особливо в контексті розробки

модулів фільтрації для систем автоматизації відділів відеоаналітики.

1.2 Класифікація модулів та методів покращення зображень для системи автоматизації відділу відеоаналітики підприємства

Класифікація модулів системи автоматизації відділу відеоаналітики підприємства може базуватися на функціональному призначенні, типах обробки даних, методах інтеграції в систему та інших аспектах. Основні категорії таких модулів наведено нижче:

1. За функціональним призначенням:

Модулі збору даних:

- Призначені для отримання даних із камер відеоспостереження чи інших пристроїв.
- Включають інтерфейси підключення до IP-камер, DVR/NVR-систем, сенсорів тощо.
- Підтримують прийом потокового відео в реальному часі або збережених записів.

Модулі попередньої обробки:

- Фільтрування шумів та артефактів у відеозображеннях.
- Нормалізація параметрів зображення, таких як яскравість, контраст, колірна гамма.
- Функції корекції спотворень через оптику або рух.

Модулі аналізу даних:

- Реалізують алгоритми розпізнавання об'єктів, обчислення траєкторій руху, визначення аномалій.
- Використовують методи машинного навчання, нейронних мереж, аналізу зображень.
- Підтримують функції класифікації об'єктів, ідентифікації осіб або номерів автомобілів.

Модулі управління:

- Організують обробку запитів користувача для отримання конкретних даних.
- Забезпечують інтеграцію з іншими системами підприємства (наприклад, ERP, системи контролю доступу).
- Підтримують функції налаштування параметрів системи.

Модулі зберігання та архівування:

- Відповідають за стиснення, зберігання та організацію доступу до відеоданих.
- Реалізують алгоритми оптимального зберігання великих обсягів даних (з використанням локальних чи хмарних сховищ).

Модулі візуалізації та звітності:

- Забезпечують інтерфейси для відображення результатів аналізу, графічних звітів, діаграм.
- Підтримують інтерактивні панелі для перегляду відео, пошуку даних у архівах, моніторингу подій у реальному часі.

2. За рівнем інтеграції в систему:

Локальні модулі

- Виконують обробку на окремому пристрої або сервері підприємства.
- Придатні для роботи в автономних системах відеоаналітики.

Мережеві модулі

- Інтегровані в локальну мережу підприємства.
- Дозволяють обробляти дані з декількох джерел одночасно та ділитися результатами.

Хмарні модулі

- Реалізовані як хмарні сервіси для масштабованої обробки та зберігання відео.
- Забезпечують доступ до даних із будь-якого місця через Інтернет.

3. За типами обробки даних:

Реального часу

- Призначені для обробки потокового відео з камер.
- Забезпечують миттєву реакцію на події (наприклад, сповіщення про проникнення).

Постобробка

- Використовуються для аналізу вже збережених відеозаписів.
- Ефективні для створення звітів, архівних перевірок або навчання алгоритмів.

4. За принципами реалізації:

Апаратні модулі

- Реалізовані у вигляді пристроїв із вбудованими процесорами для обробки відео.
- Наприклад, мережеві відеореєстратори (NVR) із функціями аналітики.

Програмні модулі

- Реалізовані у вигляді програмного забезпечення для серверів або ПК.
- Забезпечують гнучкість і можливість оновлення алгоритмів.

Комбіновані модулі

- Поєднують апаратну та програмну обробку.
- Використовують апаратне прискорення для критично важливих задач і програмні засоби для налаштувань.

5. За особливостями алгоритмів фільтрації:

Фільтри низькочастотного придушення

- Видаляють загальний шум із зображення.
- Забезпечують базову покращену якість картинки.

Адаптивні фільтри

- Автоматично налаштовуються під рівень шуму або освітлення.
- Використовуються для умов із мінливими параметрами.

Фільтри на основі машинного навчання

- Навчаються на великих наборах даних для досягнення високої точності.
- Ефективні при роботі зі складними завадами.

Модулі системи автоматизації відділу відеоаналітики підприємства поділяються на функціональні компоненти, які працюють узгоджено для вирішення задач збору, обробки, аналізу та зберігання відеоінформації. Класифікація за функціями, типами обробки та рівнями інтеграції дозволяє проектувати систему з урахуванням специфіки задач підприємства.

Основна мета покращення зображення полягає у такій його обробці, яка робить результат максимально відповідним до конкретного застосування. Наприклад, методи, ефективні для обробки рентгенівських знімків, можуть виявитися непридатними для аналізу зображень, отриманих космічними апаратами.

Існує багато підходів до покращення зображень, які поділяються на дві основні групи: методи, що працюють у просторовій області, та методи, які використовують частотну область [30]. Просторова область охоплює маніпуляції безпосередньо з пікселями зображення. Методи частотної області базуються на зміні сигналів, отриманих шляхом застосування до зображення перетворення Фур'є. Окрім того, значну роль відіграють технології, які комбінують методи обох категорій.

Методи обробки в просторовій області передбачають операції з конкретними значеннями пікселів. Процеси такої обробки можуть бути описані математичними рівняннями, які визначають зміну цих значень залежно від поставлених завдань

$$g(x, y) = T[f(x, y)], \quad (1.1)$$

де $f(x, y)$ – оригінальне зображення;

$g(x,y)$ – зображення після обробки;

T – оператор над f , заданий в деякому околі точки (x,y) .

Основний метод визначення меж точки (x, y) полягає у застосуванні квадратної або прямокутної області — підмножини зображення, яка центрована в точці (x, y) . Центр цієї області послідовно переміщується від пікселя до пікселя, починаючи з верхнього лівого кута. У кожній точці (x, y) виконується оператор T , який генерує вихідне значення g для даної точки. Обробка здійснюється лише з використанням пікселів, що знаходяться в межах обраної області зображення. Найчастіше застосовують квадратні або прямокутні області через їхню простоту реалізації, хоча іноді використовуються області іншої форми, наприклад, наближені до кола. Найбільш простим випадком оператора T є ситуація, коли область має розмір 1×1 (один піксель). У такому разі g залежить тільки від значення f у точці (x, y) , і оператор T стає функцією градаційного перетворення, відомою також як «функція перетворення інтенсивності» або «функція відображення»

$$s = T(r), \quad (1.2)$$

де r і s – параметри, які визначають відповідні значення яскравостей зображень $f(x,y)$ та $g(x,y)$ в кожній точці (x,y) .

Оскільки покращення якості кожного пікселя зображення визначається його яскравістю, методи цієї групи зазвичай відносять до поелементних процедур обробки. Збільшення розміру околу забезпечує значно більшу гнучкість. Основна ідея полягає в тому, що для обчислення значення g у конкретній точці (x, y) використовуються значення функції f , розташовані в межах певного околу заданої форми, який знаходиться поблизу цієї точки. Один із ключових підходів у цьому процесі базується на використанні так званих масок (інша назва — фільтри, ядра, шаблони або вікна). Як правило, маска є невеликим двовимірним масивом (найчастіше розміром 3×3), де

коефіцієнти задають характер обробки, наприклад, для підвищення різкості зображення. Методи, які спираються на цей принцип, часто класифікують як обробку за допомогою масок або фільтрацію маскою [30].

Процедура фільтрації в частотній області проста і містить означені кроки:

- вихідне зображення множиться на $(-1)^{x+y}$, з метою щоб його Фур'є перетворення виявилось центрованим;
 - обчислюється $F(u,v)$ – пряме перетворення Фур'є зображення, що отримане після першого кроку;
 - відбувається множення функція $F(u,v)$ на функцію фільтра $H(u,v)$;
 - обчислюється обернене перетворення Фур'є від результату третього кроку;
 - виділяється дійсна частина результату четвертого кроку;
- результат п'ятого кроку множиться на $(-1)^{x+y}$.

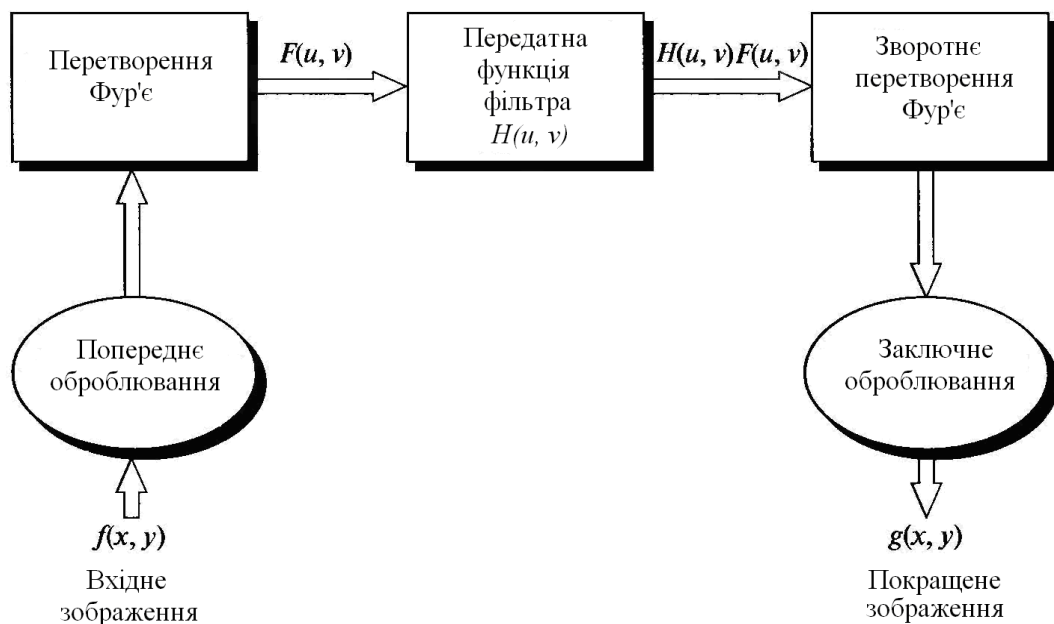


Рисунок 1.1 – Ключові етапи фільтрації у частотній області

Багато прикладів демонструють, що деякі завдання покращення зображень, які є надзвичайно складними або навіть неможливими для вирішення у

просторовій області, значно спрощуються при переході до частотної області. Після експериментального підбору відповідного фільтра в частотній області його реалізація зазвичай здійснюється вже у просторовій області.

Основний зв'язок між фільтрацією в просторовій і частотній областях визначається відомою «теоремою про згортку». Суть операції згортки полягає в тому, що маска, пересуваючись по зображенню, для кожного пікселя обчислює визначену величину за наперед заданими правилами.

1.3 Аналіз методів фільтрації цифрових зображень

Методи фільтрації зображень класифіковано на рисунку 1.2.

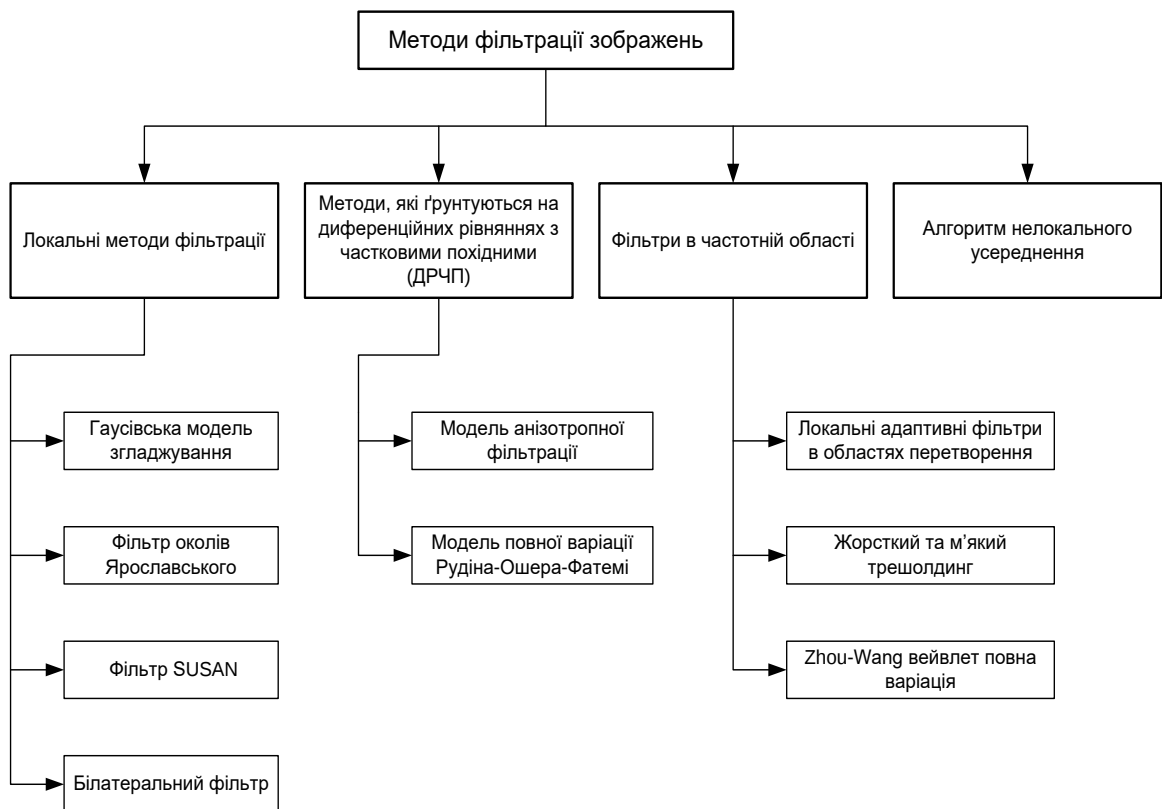


Рисунок 1.2 – Класифікація методів фільтрації зображень

Зображення, отримане за допомогою технічних пристроїв, можна математично подати у такій формі:

$$v(i) = u(i) + n(i) \quad (1.3)$$

де $v(i)$ – величина, яка спостерігається;

$u(i)$ – дійсна величина, яку необхідно відновити з $v(i)$;

$n(i)$ – шумове відхилення.

Метод очищення від шуму зображення u може бути означений як D_h :

$$u = D_h u + n(D_h, u), \quad (1.4)$$

де h – параметр фільтрації;

$D_h u$ – очищене від шуму зображення;

$n(D_h, u)$ – шум, визначений цим методом.

На сьогоднішній день недостатньо просто згладити зображення, щоб отримати очищене зображення. Сучасні методи не лише виконують згладжування, але й можуть відновлювати втрачений контент, якщо це необхідно [79]. Наприклад, у зображеннях, отриманих за допомогою цифрових дзеркальних камер, часто постає завдання зберегти чіткість і деталі, при цьому необхідно зменшити або розмити шум, що може виникнути під час зйомки.

1.3.1 Локальні методи фільтрації

Оригінальне (півтонове) зображення u визначається в обмеженій області $\Omega \subset R^2$ та позначається як $u(x)$ для $x \in R^2$. Згідно з

загальноприйнятою практикою візуалізації на екрані або принтері, дискретні значення шуму інтерполюються не як функція з обмеженим спектром, а як кусочно-постійна функція, де значення залишаються постійними в межах кожного пікселя i рівні.

Можна записати $|x|$ як L^2 норму та $x \cdot u$ як скалярний добуток.

Гаусівський фільтр є одним із найпоширеніших інструментів для розмиття зображень, який працює шляхом згортки зображення з лінійним симетричним ядром. Процес згладжування зазвичай базується на використанні позитивних значень ядра. Це ядро, відоме як ядро Гауса, формується на основі математичного виразу функції Гауса. Завдяки своїм властивостям, фільтр ефективно зменшує високочастотні компоненти, такі як шум, зберігаючи при цьому основні структурні елементи зображення

$$x \rightarrow G_h(x) = \frac{1}{(4\pi h^2)} e^{-\frac{|x|^2}{4h^2}} \quad (1.5)$$

Функція G_h має середнє квадратичне відхилення. Обчислення є правильним, якщо значення цього відхилення h є достатньо малим. Ефективність зменшення шуму значною мірою залежить від використання достатньо великих околів пікселів під час згладжування. Це дозволяє виконати процес осереднення, у результаті якого значення пікселів, що відповідають шумовим компонентам, поступово "згладжуються", знижуючи їхній вплив на зображення. Такий підхід не лише зменшує шум, але й сприяє збереженню ключових деталей, що покращує загальну якість зображення. Якщо припустити, що $h = k\epsilon$, де k – кількість відрахунків функції u та шуму в інтервалі довжиною h , ϵ^2 – розмір локального вікна, k має бути більше за 1.

В пікселі $i = 0$ гаусівський ефект згладження обчислюється наступним чином:

$$G_h * n(0) = \sum_{i \in I} \int_{P_i} G_h(x) n(x) dx = \sum_{i \in I} \epsilon^2 G_h(i) n_i \quad (1.6)$$

де $n(x)$ – функція, інтерпольована як кускова;

P_i – квадратні пікселі з центром у i та розміром ϵ^2 ;

$G_h(i)$ – середнє значення функції G_h у пікселі i [20].

Фільтри околів використовують значення рівнів сірого для визначення сусідніх пікселів. В цьому випадку, очищене від шуму значення пікселя i обчислюється як зважене середнє значень пікселів, рівні сірого яких найбільш схожі на рівень сірого пікселя $u(i)$. Окіл рівня сірого можна визначити як

$$U(i, h) = \{j \in I | u(i) - h < u(j) < u(i) + h\}. \quad (1.7)$$

Цей метод можна вважати локальним у межах інтенсивності (яскравості), оскільки обробка базується на значеннях пікселів. Водночас він є нелокальним у просторовій області, оскільки для обчислення значення пікселя в певній точці враховується інформація з інших частин зображення. Такий підхід дозволяє враховувати взаємодії між пікселями, розташованими у різних ділянках зображення, що сприяє більш точному зменшенню шуму та збереженню важливих деталей. *і*. Популярним варіантом цього алгоритму є середньозважений фільтр

$$NH_h u(x) = \frac{1}{C(x)} \int_{\Omega} u(y) e^{-\frac{|u(y)-u(x)|^2}{h^2}} dy \quad (1.8)$$

де $\Omega \subset R^2$ є відкритою та обмеженою множиною;

$$C(x) = \int_{\Omega} u(y) e^{-\frac{|u(y)-u(x)|^2}{h^2}} dy \quad \text{— коефіцієнт нормалізації.}$$

Фільтри околів [5] розглядають змішані околи $U(i, h) \cap B_{\rho}(i)$, де $B_{\rho}(i)$ — коло з центром i та радіусом ρ .

Фільтр SUSAN (Smallest Univalued Segment Assimilating Nucleus — найменший однорідний сегмент, який асимілюється ядром):

$$YNF_{h,p}(x) = \frac{1}{C(x)} \int_{B_p(x)} u(y) e^{-\frac{|u(y)-u(x)|^2}{h^2}} dy \quad (1.9)$$

де $C(x)$ — коефіцієнт нормалізації [82].

Білатеральний фільтр (БФ), запропонований у 1998 році в [3], є методом, який забезпечує згладжування зображення без розмивання меж. Це досягається завдяки комбінації двох гаусівських фільтрів: один із них діє в просторовій області, а інший — в області інтенсивності (яскравості). Такий підхід дозволяє враховувати не лише просторову відстань між пікселями, а й різницю в їх інтенсивності, що є важливим для точного визначення вагових коефіцієнтів. Завдяки цьому білатеральний фільтр зберігає деталі на межах об'єктів, уникнувши їх розмиття, що є важливим для збереження чіткості зображення при одночасному зменшенні шуму.

Для $u(x)$ у пікселі x , білатеральний фільтр може бути знайдений як

$$B(x) = \frac{1}{C(x)} \sum_{y \in N(x)} e^{-\frac{\|y-x\|^2}{2\sigma_d^2}} e^{-\frac{|u(y)-u(x)|^2}{2\sigma_r^2}} u(y) \quad (1.10)$$

де σ_d та σ_r – параметри регулювання зменшення ваг у просторовій (відстань) області та в області інтенсивності (радіометрична);

$N(x)$ – просторовий окіл пікселя $u(x)$, та коефіцієнт нормалізації.

$$C(x) = \sum_{y \in N(x)} e^{-\frac{\|y-x\|^2}{2\sigma_d^2}} e^{-\frac{|u(y)-u(x)|^2}{2\sigma_r^2}} \quad (1.11)$$

Білатеральний фільтр є популярним методом для зменшення шуму в зображеннях, однак відсутнє чітке теоретичне обґрунтування для вибору оптимального розміру вікна околу. Зазвичай цей параметр визначається експериментально. Очевидно, що якщо розмір вікна занадто малий, ефективність фільтрації шуму обмежена, оскільки враховується недостатньо пікселів, що мають взаємозв'язок. З іншого боку, збільшення розміру вікна, тобто використання більш глобального околу, може включати в обробку більше непов'язаних пікселів, що негативно впливає на якість знешумлення. При надмірно великому розмірі вікна фільтр стає менш ефективним, оскільки алгоритм може включати надмірну кількість пікселів, які не допомагають у правильному зменшенні шуму [24].

1.3.2 Методи, що ґрунтуються на диференціальних рівняннях з частковими похідними

Дифузійні алгоритми є потужним інструментом для зменшення шуму в зображеннях, зберігаючи при цьому важливі контури. Вони засновані на диференціальному рівнянні з частковими похідними (ДРЧП), яке моделює процес дифузії, де зображення проходить через кілька етапів фільтрації. Цей підхід дозволяє адаптувати фільтрацію до основного сигналу зображення на локальному рівні, зберігаючи деталі та контури, водночас ефективно усуваючи шум. Основною перевагою дифузійних алгоритмів є їх здатність зберігати структуру зображення, обробляючи лише шуми, без значних змін у важливих характеристиках зображення.

Припустимо, що $I(x, t)$ – зображення в дифузійному процесі з певним часом t , $I(x, U): R^2 \rightarrow R^+$ – оригінальне зображення в безперервній області, $x \in R^2$ – положення пікселя в 2D зображенні. Ізотропне дифузійне рівняння

$$\frac{\partial I(x, t)}{\partial t} = \text{div}(\nabla I), \quad I(x, 0) = u \quad (1.12)$$

застосовувалося для зменшення шуму на зображеннях. Дослідження показали, що модифікація зображення за допомогою цього анізотропного дифузійного рівняння еквівалентна процесу фільтрації з використанням фільтра Гаусса [5].

Мінімізація повної варіації (TV) була запропонована у [6]. Нехай $v(x) = u(x) + n(x)$, де u позначає оригінальне зображення без шуму, а n — шум. Для відновлення ставиться задача мінімізації

$$E(u) = \lambda \|u - v\|_{L^2(\Omega)}^2 + R(u) \quad (1.13)$$

де $\lambda > 0$, Ω – область на якій v визначається;

$R(u)$ – оператор регуляризації.

Раніше увага була зосереджена на операторі $R(u)$, який базувався на методі найменших квадратів, таких як $\|\Delta u\|_{L^2(\Omega)}^2$, $\|\nabla u\|_{L^2(\Omega)}^2$ та інших. Застосування оператора регуляризації разом з видаленням шуму призводило до отримання зображень з меншою чіткістю та згладженими деталями, що спричиняло втрату важливих елементів.

Оптимальний вибір для функціонального просторового моделювання цих характеристик $BV(\Omega)$, обмеженою повною варіацією, що підлягають інтегруванню $TV_\Omega(u) = \int |\nabla u|$. Для зображення $v(x)$ з шумом, запропоновано відновити зображення $u(x)$ як розв'язок обмеженої задачі мінімізації

$$\arg \min_u TV_\Omega(u) \quad (1.14)$$

за умови обмеження шуму

$$\int_{\Omega} (u(x) - v(x)) dx = 0 \text{ та } \int_{\Omega} |u(x) - v(x)|^2 dx = \sigma^2 \quad (1.15)$$

Рішення u має бути настільки точним, наскільки це можливо у плані повної варіації, в той час як різниця $u - v$ розглядається як помилка з заданими енергетичними значеннями. Обмеження визначає вірне середнє значення та дисперсію для $u - v$, однак не забезпечує, що воно буде подібним до шуму [87]. Таким чином, з попередньої задачі

$$\arg \min_u \left(TV_{\Omega}(u) + \lambda \int_{\Omega} |v(x) - u(x)|^2 dx \right) \quad (1.16)$$

Наведена вище функція є строго опуклою та напівбезперервною знизу щодо топології слабкої зірки. Тому існуючий мінімум є єдиним і обчислюваним [8]. Параметр визначає співвідношення між точністю та правильністю границі. Таким чином, цей параметр пов'язаний із рівнем фільтрації при розв'язуванні задачі мінімізації. Дослідження показують, що повна варіація краще зберігає границі, що дозволяє здійснювати ефективнішу реконструкцію [8-11]. Серед усіх основних методів ДРЧП схема мінімізації забезпечує найкраще поєднання видалення шуму та збереження властивостей.

Під час подальшого дослідження видалення шуму в початковій моделі повної варіації у [12] була запропонована нова ітераційна модель повної варіації:

1. Вирішити

$$u_1 = \arg \min_{u \in BV} \left\{ \int |\nabla u(x)| dx + \lambda \int (v(x) - u(x))^2 dx \right\} \quad (1.17)$$

2. Провести крок корекції

$$u_2 = \arg \min_{u \in BV} \left\{ \int |\nabla u(x)| dx + \lambda \int (u_1(x) - u(x))^2 dx \right\} \quad (1.18)$$

де u_1 перетворюється на нове зображення, яке спостерігається.

3. Провести ітерацію: обчислити u_{k+1} як мінімум модифікованої мінімізації повної варіації

$$u_{k+1} = \arg \min_{u \in BV} \left\{ \int |\nabla u(x)| dx + \lambda \int (u_k(x) - u(x))^2 dx \right\} \quad (1.19)$$

В [12] доведено, що:

- u_k сходиться монотонно у L^2 до v , зображення з шумом при $k \rightarrow \infty$.
- u_k наближається до зображення без шуму монотонно у відстані Брегмана, пов'язаній з напівнормою BV, принаймні доки $\|u_k - u\| \leq \sigma^2$, де u початкове зображення, а σ – типове (стабільне, середньоквадратичне) відхилення доданого шуму.

Отримані результати дозволяють припинити послідовність та обрати u_k . Цього достатньо щоб продовжити ітерації поки результат не стане менш зашумленим або поки відстань не досягне заданого порогу $\|u_k - u\|^2$ та стане меншою за σ^2 . Таке рішення дозволяє зберегти більше деталей.

1.3.3 Фільтри в частотній області

Припустимо, що u - зображення, яке визначене на R^n . Припускається, що модифікація зображення білим шумом представлена як випадковий процес. Тоді, зображення v , яке спостерігається може бути записне наступним чином:

$$v(i) = u(i) + n(i)$$

Нехай $B = \{g_\alpha\}_{\alpha \in A}$ буде ортогональним базисом R^n .

$$v_B(\alpha) = \langle v, g_\alpha \rangle, \quad u_B(\alpha) = \langle u, g_\alpha \rangle, \quad n_B(\alpha) = \langle n, g_\alpha \rangle$$

визначається як скалярні добутки. Тоді

$$v_B(\alpha) = u_B(\alpha) + n_B(\alpha)$$

стане трансформованим процесом шуму.

Параметри шуму $n_B(\alpha)$ лишаються некорельованими і дорівнюють нульовим середнім значенням, при цьому дисперсії множиться на $\|g_\alpha\|^2$

$$\begin{aligned} E[n_B(\alpha)n_B(\beta)] &= \sum_{i,j \in I} g_\alpha(i)g_\beta(j)E[n(i)n(j)] = \\ &= \langle g_\alpha, g_\beta \rangle \sigma^2 = \sigma^2 \|g_\alpha\|^2 \delta[\alpha - \beta] \end{aligned} \quad (1.20)$$

Частотні фільтри діють окремо на кожен трансформований коефіцієнт $v_B(\alpha)$. Для відновлення зображення виконується обернене перетворення нових коефіцієнтів. Параметри шуму $v_B(\alpha)$ перетворюються на $a(\alpha)v_B(\alpha)$, де $a(\alpha)$ часто обмежуються значенням 0 або 1. Алгоритм нелінійний тому, що $a(\alpha)$ залежить від значення $v_B(\alpha)$. Обернене перетворення дає оцінку

$$\hat{U} = DV = \sum_{\alpha \in A} a(\alpha) v_B(\alpha) g_\alpha \quad (1.21)$$

Класичний частотний фільтр діапазону, що використовує базис Фур'є, може призвести до того, що глобальні характеристики зображення домінують над локальними, утворюючи помилкові періодичні візерунки. Щоб уникнути цього, потрібно використовувати базиси, які краще враховують локальні особливості зображення, такі як вейвлет-перетворення та локальні перетворення ДКП. Визначення оптимального базису для кожного окремого випадку залишається ключовою задачею в обробці зображень.

Методи вейвлет-трешолдингу були вперше запропоновані Д. Донохо [13]. Припустимо, що $B = \{g_\alpha\}_{\alpha \in A}$ - ортогональний вейвлет базис [16]. Для визначення $a(\alpha)$ обирається певний поріг μ , тобто

$$a(\alpha) = \begin{cases} 1 & |v_B(\alpha)| > \mu \\ 0 & |v_B(\alpha)| \leq \mu \end{cases} \quad (1.22)$$

Цей метод відомий як жорстке обмеження або жорсткий трешолдинг, що полягає у відкиданні коефіцієнтів, які є меншими за встановлений поріг. Підхід базується на припущенні, що ключові елементи зображення визначаються великими вейвлет-коефіцієнтами, котрі зберігаються, в той час як шум, як правило, представлений малими коефіцієнтами, що обнуляються. Ефективність методу визначається здатністю передавати апроксимацію шляхом використання обмеженої кількості великих коефіцієнтів. Позначимо цей оператор як $HWT_\mu(v)$.

Однак, через просте обнулення вейвлет-коефіцієнтів, які знаходяться нижче порогу, цей алгоритм може спричиняти появу невеликих коливань, що призводять до явища Гіббса поблизу границь. Донохо [14] довів, що ці ефекти можуть бути частково зменшені при використанні м'якого граничного оброблення (м'якого трешолдингу)

$$a(\alpha) = \begin{cases} \frac{v_B(\alpha) - \text{sgn}(v_B(\alpha))\mu}{v_B(\alpha)} & |v_B(\alpha)| > \mu \\ 0 & |v_B(\alpha)| \leq \mu \end{cases}, \quad (1.23)$$

яке позначимо як $SWT_\mu(v)$. Оператор м'якого граничного оброблення, завдяки своїй неперервності, зберігає структуру вейвлет-коефіцієнтів і мінімізує коливання на межах зображення.

У 2006 році в [5] був представлений алгоритм вейвлет-повної варіації для обробки медичних зображень, який поєднує мінімізацію повної варіації з вейвлет-підходом [6, 7]. На відміну від вейвлет-трешолдингу жорсткого та м'якого типу, цей алгоритм дозволяє уникати ефекту Гіббса. Однією з основних проблем традиційної моделі є надмірне згладжування. Для вейвлет-моделі Ванг і Чжоу розробили ефективний автоматичний критерій зупинки. В порівнянні з традиційними підходами, застосування цього критерію зупинки, що базується на статистичних властивостях вейвлет-коефіцієнтів, дозволяє уникнути надмірного згладжування зображення [5].

1.4 Висновки до першого розділу

При отриманні зображень в умовах низької освітленості, наприклад, вночі, їх якість значно погіршується через збільшення рівня шуму, що призводить до появи зернистості. Підсилення сигналу в таких умовах викликає додатковий шум, який негативно впливає на ефективність стиснення зображень. Сучасні алгоритми компресії часто сприймають шум як корисну інформацію, що знижує ефективність стиснення. Внаслідок цього обсяг переданої інформації збільшується, тоді як корисна інформація зменшується. Аналіз наявних методів фільтрації показав, що більшість з них

орієнтовані на обробку зображень з півтоновими градаціями і штучно доданим шумом. Лише деякі підходи адаптовані для роботи з природними кольоровими зображеннями, отриманими за допомогою цифрових камер, що містять реальний шум. Проте ці методи часто є або дуже складними, або не дають достатньо ефективних результатів у зменшенні шуму. У контексті розробки модуля фільтрації для системи автоматизації відеоаналітики підприємства важливою є задача покращення співвідношення сигнал/шум, що включає необхідність попередньої обробки зображень перед їх класифікацією та аналізом, особливо в процесах розпізнавання образів і прийняття рішень.

2 РОЗРОБКА МЕТОДУ ФІЛЬТРАЦІЇ НА ОСНОВІ БІЛАТЕРАЛЬНОГО ФІЛЬТРУ

Розробка методу фільтрації для реалізації модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства ґрунтується на кількох ключових аспектах:

1. **Покращення якості зображень у реальному часі:** У процесі відеоаналітики на підприємстві, зображення з камер спостереження часто містять шум, що виникає через низьку освітленість, погану фокусування камери або інші технічні обмеження. Для ефективного аналізу та розпізнавання об'єктів на таких зображеннях необхідно зменшити рівень шуму, не втрачаючи важливих деталей. Розробка ефективного методу фільтрації допомагає отримати більш чітке зображення, що є критичним для автоматичних систем моніторингу та прийняття рішень.
2. **Зменшення впливу шуму на алгоритми аналізу:** Шум на зображеннях може суттєво знижувати ефективність алгоритмів розпізнавання образів, класифікації та детекції об'єктів. Наприклад, у випадку з відеоаналітикою на підприємстві, неправильне розпізнавання об'єктів або подій через шум може призвести до помилок в системі моніторингу. Вибір методу фільтрації, який оптимально знижує шум без втрати важливих характеристик зображення, є необхідним для досягнення високої точності в роботі таких систем.
3. **Збереження важливих деталей і контурів:** Багато існуючих методів фільтрації, хоча й ефективні для усунення шуму, можуть розмивати важливі деталі зображень, зокрема контури об'єктів. Для відеоаналітики на підприємствах важливо зберігати чіткість важливих елементів, таких як контури людей, транспортних засобів, об'єктів, що рухаються. Тому розробка методу фільтрації повинна орієнтуватися на

баланс між зменшенням шуму і збереженням критично важливих деталей.

4. Оптимізація для реального часу: Відеоаналітика на підприємствах часто вимагає обробки зображень в реальному часі. Тому важливо розробити метод фільтрації, який забезпечить зниження шуму з мінімальними затримками. Це дозволить системі миттєво реагувати на події, що відбуваються, і забезпечить ефективну автоматизацію.
5. Адаптивність до різних умов зйомки: У реальних умовах зйомки, наприклад, у складних освітлювальних умовах, камера може бути схильна до різного рівня шуму. Тому важливим аспектом є адаптивність методу фільтрації до різних рівнів шуму та змін у якості зображень, що дозволить системі бути ефективною в будь-яких умовах.

Таким чином, розробка методу фільтрації для модуля фільтрації цифрових зображень є важливим етапом для підвищення ефективності відеоаналітики на підприємстві. Це дозволяє забезпечити точність, швидкість і надійність систем автоматизації в обробці зображень і прийнятті рішень на основі відеоданих.

2.1 Дискретне вейвлет-перетворення

Дискретне вейвлет-перетворення (ДВП) надає багатошкальне представлення сигналу, яке ґрунтується на ідеї обережного виявлення границь через використання вейвлет-теорії. Однією з головних переваг ДВП є його трансляційна інваріантність, що означає, що перетворення не залежить від переміщення сигналу. Це дозволяє ефективно аналізувати сигнал на різних рівнях деталізації. Крім того, ДВП здатне характеризувати локальні особливості нерегулярних структур, що досягається шляхом виявлення локальних максимумів вейвлетів для звичайних сигналів. Завдяки цьому,

вейвлет-перетворення є потужним інструментом для аналізу зображень та сигналів, що мають різну ступінь регулярності і складності [4].

Нехай $\Psi(x)$ – вейвлет функція та $f(x) \in L^2(R)$. Дилатацію $\Psi(x)$ з показником 2^j можна визначити як [4]:

$$\psi_{2^j}(x) = \frac{1}{2^j} \psi\left(\frac{x}{2^j}\right). \quad (2.1)$$

Вейвлет перетворення для $f(x)$ у масштабі 2^j та позиції x визначається згорткою [4]:

$$W_{2^j} f(u) = \left(f(x) * \psi_{2^j}(x) \right)(u) = \int_{-\infty}^{\infty} f(x) \frac{1}{\sqrt{2^j}} \psi\left(\frac{x-u}{2^j}\right) dx. \quad (2.2)$$

Це вказує на те, що масштаб для неперервного вейвлет-перетворення є дискретним уздовж діадної послідовності $\{2^j\}$, причому j належить цілочисленній множині Z .

Діадне вейвлет-перетворення є набором функцій [24]:

$$Wf = \left(W_{2^j} f(x) \right)_{j \in Z}, \quad (2.3)$$

де W – оператор діадного вейвлет-перетворення.

Припустимо що функція $\phi(x)$ згладжувальна функція відповідна вейвлет-функції $\Psi(x)$ та S_{2^j} згладжувальний оператор, який визначається як згортка:

$$S_{2^j} f(u) = \left(f(x) * \phi_{2^j}(x) \right)(u) = \int_{-\infty}^{\infty} f(x) \frac{1}{2^j} \phi\left(\frac{x-u}{2^j}\right) dx, \quad (2.4)$$

причому $\phi_{2^j}(x)$ визначений як

$$\phi_{2^j}(x) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{x}{2^j}\right), \quad (2.5)$$

тоді, з вищезгаданого визначення, можна відмітити, що більше деталей $f(x)$ видалені S_{2^j} , коли масштаб 2^j стає більшим.

Якщо оригінальний сигнал є дискретною послідовністю $f = (f(n))_{n \in \mathbb{Z}}$ з обмеженою енергією, можна припустити, що фрагменти $a_0(n)$ вхідного дискретного сигналу не точно дорівнюють $f(n)$, але локальне середнє f по сусідству $t = n$. Таким чином, $a_0(n)$ можна виразити наступною формулою

$$a_0(n) = \langle f(t), \phi(t-n) \rangle = \int_{-\infty}^{\infty} f(t) \phi(t-n) dt. \quad (2.6)$$

Для кожного $j > 0$ можна відмітити

$$S_{2^j} f = a_j(n) = \langle f(t), \phi_{2^j}(t-n) \rangle = \int_{-\infty}^{\infty} f(t) \phi_{2^j}(t-n) dt. \quad (2.7)$$

Діадне вейвлет-перетворення

$$W_{2^j} f = d_j(n) = Wf(n, 2^j) = \langle f(t), \psi_{2^j}(t-n) \rangle. \quad (2.8)$$

Для кожного масштабу 2^J послідовність дискретних сигналів $\left(S_{2^J} f, (W_{2^J} f)_{1 \leq j \leq J} \right)$ є дискретним діадним вейвлет-перетворенням $f = (f(n))$.

Отже, можна визначити діадне вейвлет-перетворення $(S_2 f, W_2 f)$ як (Sf, Wf) для функції $f(n)$, де Sf - низькочастотна інформація $f(n)$, а Wf - високочастотна інформація $f(n)$.

2.2 Опис методу обробки цифрових зображень за допомогою білатерального фільтра

Чимало досліджень присвячено застосуванню білатерального фільтра. Цей фільтр був адаптований для вирішення задач, таких як видалення текстури [7], динамічне стиснення на основі рангу [8] та обробка зображень [9, 10]. Він також знайшов успішне застосування в інших сферах, зокрема для виділення контурів [11, 12], зменшення об'ємного шуму [13] і корекції відео. Така універсальність білатерального фільтра пояснюється кількома його перевагами. По-перше, його легко формалізувати та реалізувати, оскільки кожен піксель замінюється на зважене середнє значення сусідніх пікселів. По-друге, білатеральний фільтр не вимагає ітерацій, що дозволяє отримати результат після одного проходу, і параметри фільтра є інтуїтивно зрозумілими, оскільки їх дія не залежить від накопичення ефектів кількох ітерацій. З іншого боку, фільтр є нелінійним, що обмежує його застосування без додаткових удосконалень і вимагає великих обчислювальних ресурсів. Результатом роботи білатерального фільтра є розрахунок інтенсивності кожного пікселя як зваженого середнього значення інтенсивностей його сусідів [13]. Вага кожного сусіднього пікселя визначається з урахуванням як просторової відстані (просторова вага), так і різниці інтенсивностей (рангова

вага), що дозволяє зберігати чіткість меж об'єктів на зображенні. Використовуючи функцію Гауса G_σ у вигляді залежності і вважаючи вхідне зображення I чорно-білим, визначимо результат білатерального фільтра для кожного пікселя p :

$$I_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (2.9)$$

$$W_p = \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|), \quad (2.10)$$

де: S – множина можливих позицій пікселів зображення;

$p = (p_x, p_y) \in S$ – позиція центрального пікселя;

q – позиція сусіднього пікселя;

I_p, I_q – інтенсивності пекселів p і q в діапазоні $[0, 1]$;

$G_{\sigma_d}, G_{\sigma_r}$ – функції Гауса з параметрами σ_d і σ_r ;

W_p – сума вагів для нормалізації значень.

Параметр σ_d задає число сусідніх пікселів, значення яких враховується при обробці одного пікселя області S , σ_r відображає втрату у вазі сусідніми пікселями при різниці інтенсивностей. На рисунку 2.1 показано процес обчислення ваг для окремого пікселя на прикладі простого двовимірного сигналу.

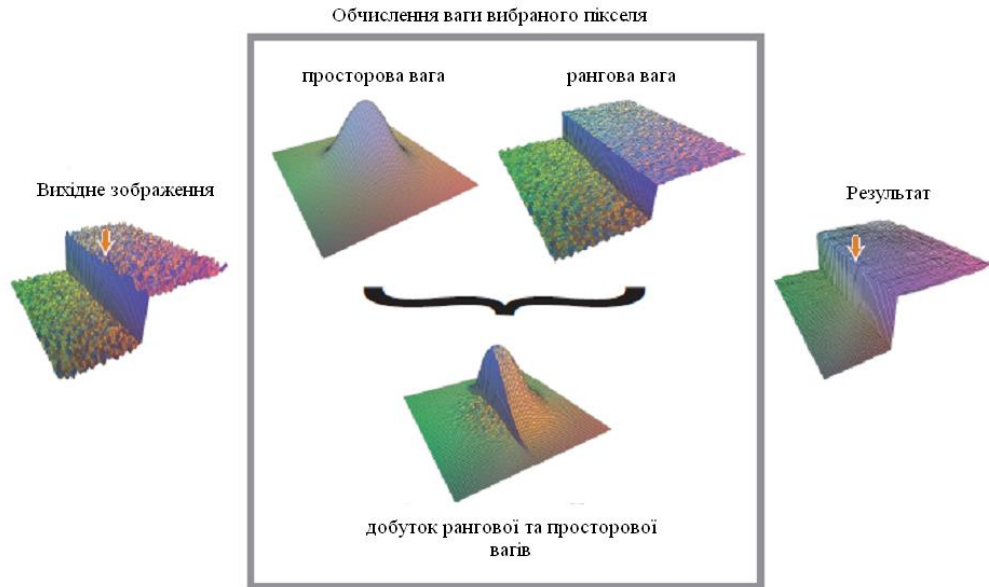


Рисунок 2.1 – Застосування білатерального фільтра для згладжування вихідного сигналу з урахуванням збереження контурів

Білатеральний фільтр можна також реалізувати за допомогою швидкого перетворення Фур'є [8], однак Гаусів білатеральний фільтр здобув більше практичне застосування. Однак точна обробка за допомогою цього методу вимагає значних обчислювальних ресурсів, що робить його непрактичним, зокрема для попередньої обробки цифрових даних. Існує чотири основних підходи для наближеного застосування білатеральної фільтрації [3].

При реалізації білатеральної фільтрації прямим методом обчислення інтенсивності пікселя відбувається по обмеженій області $B_{\sigma_d}^2$ за наступною формулою:

$$I_p = \frac{1}{W_p} \sum_{q \in B_{\sigma_d}} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (2.11)$$

Ця область охоплює пікселі, що знаходяться на відстані не більше ніж вдвічі перевищує стандартне відхилення від центрального пікселя $2\sigma_d$, тому що вплив решти пікселів незначний. Складність алгоритму за даним методом $O(|S|\sigma_d^2)$. Таке наближення ефективно тільки для невеликих ядер (малих значеннях параметра σ_d) через квадратичну залежності від σ_d . Використовуючи цей алгоритм, можна досягти результату, що максимально наближений до точного, однак це потребує значних часових витрат. Наприклад, обробка зображення розміром 8 Мп може займати більше 10 хвилин.

За методом ядра двовимірний фільтр розділяється на два одновимірних, які застосовуються по черзі. Спочатку зображення фільтрується по стовпцях, а потім — по рядках. Для кожного етапу використовується стандартний метод білатеральної фільтрації, адаптований для одновимірних масивів результатів. Така стратегія значно скорочує час обчислень, оскільки кількість пікселів, що обробляються, залежить від радіусу ядра. Складність даного алгоритму стає рівною $O(|S|\sigma_d)$, тому що множина розглянутих пікселів одновимірна. Час обчислень за цим алгоритмом значно менший порівняно з прямим методом і зростає лінійно при збільшенні розміру ядра.

Цей метод фільтрації показує високу ефективність при використанні невеликих ядер і може бути застосований для обробки відео через багаторазове застосування фільтрації з малими ядрами. Однак він не підходить для великих ядер (деяких пікселів), оскільки може виникати видимі артефакти. Зі збільшенням розміру ядра область, яка використовується для розрахунку ваг кожного пікселя, охоплює все більше різноманітних елементів зображення, таких як різні краї об'єкта і його внутрішні межі. Проте ці деталі не впливають на результати фільтрації завдяки послідовному обробленню рядків і стовпців. Існує також

вдосконалена версія цього методу, де розподіл проводиться по осях, орієнтованих згідно з градієнтами зображення. Хоча цей підхід потребує попереднього аналізу зображення, він дозволяє значно поліпшити якість результату. Метод локальних гістограм не враховує відстань між пікселями.

Результат обчислюється по пікселях, які захоплюються квадратним ядром $N\sigma_d$ (локальною гістограмою) за наступною формулою:

$$I_p = \frac{1}{W_p} \sum_{q \in N\sigma_r} G_{\sigma_r}(|I_p - I_q|) I_q \quad (2.12)$$

де $N_{\sigma_r}(p) = \{q, \|p - q\| \leq \sigma_s\}$.

Прискорення обчислень досягається завдяки виключенню просторової ваги, що призводить до того, що локальні гістограми для двох сусідніх пікселів майже повністю збігаються. Складність алгоритму обчислюється як логарифм від радіуса ядра $O(|S| \log \sigma_d)$. Отже, час виконання цього методу зазвичай складає кілька секунд навіть для великих ядер (радіусом до 100 пікселів для зображень з роздільною здатністю понад 8 Мп).

Основною проблемою цього методу є поява ореолів навколо різких меж, що відомі як смуги Маху. Для зменшення цього ефекту можна використовувати фільтрацію в три етапи, зберігаючи постійними рангові ваги, що дозволяє зменшити артефакти на межах. Проте таке вдосконалення ускладнює налаштування алгоритму, особливо для кольорових зображень і завдань з великим обсягом оброблюваних даних. Ідея методу білатеральної решітки полягає в тому, щоб результат фільтрації залежав лише від відстані між пікселями. Зображення відображається в тривимірному просторі, де до стандартних координат додається координата інтенсивності пікселя, що дозволяє обчислити однорідний вектор за відповідною формулою:

$$\begin{pmatrix} I_p \cdot W_p \\ W_p \end{pmatrix} = \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) \begin{pmatrix} I_q \\ 1 \end{pmatrix} \quad (2.13)$$

Отриманий простір стискається з певною частотою дискретизації, де кожен піксель зберігає свою інформацію в відповідній накопичувальній клітинці. Така структура, що складається з таких осередків, отримала назву білатеральної решітки. Потім зображення обробляється за допомогою білатерального фільтра, який враховує лише відстань між пікселями. Зокрема, проводиться згортка з тривимірною функцією Гауса з відповідними

параметрами $\frac{\sigma_d / S_d, \sigma_r}{S_r}$, що дозволяє зберігати важливі риси зображення при зменшенні впливу шуму і артефактів, після цього результат відновлюється відповідно до початкових позицій пікселів за допомогою трилінійної інтерполяції та нормалізації. Цей алгоритм дозволяє обробляти зображення з ядрами будь-якого розміру за короткий час (приблизно 1 секунду).

Складність представленого алгоритму $O\left(|S| + \frac{|S| |R|}{\sigma_d^2 \sigma_r}\right)$, де $|S|$ – розмір зображення в пікселях, $|R|$ – діапазон шкали інтенсивностей.

Основною перевагою методу швидкої білатеральної фільтрації є його гнучкість та здатність до адаптації для різних завдань і розширень. Це дозволяє використовувати метод для обробки кольорових зображень або навіть для одночасної роботи з кількома зображеннями. Як було зазначено в першому розділі, шум на зображеннях не завжди є «білим» і може мати різні просторово-частотні властивості, такі як дрібнозернистий чи крупнозернистий. Завдяки аналізу на кількох масштабах можна точніше розрізняти шум і корисну інформацію на кожному рівні розкладу, що

дозволяє ефективніше застосовувати білатеральний фільтр в кратномасштабній структурі для покращення фільтрації (рис. 2.2).

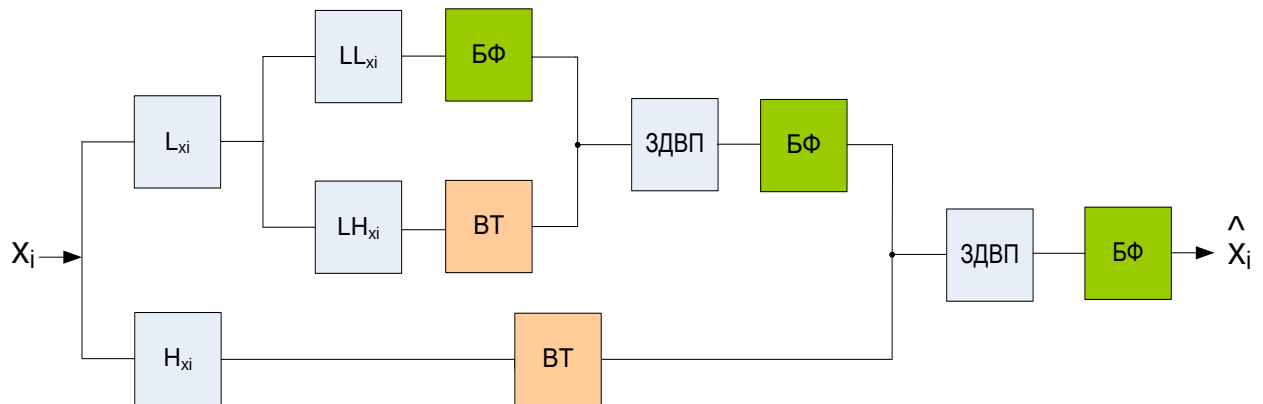


Рисунок 2.2 – Процес фільтрації за допомогою білатерального фільтра в кратномасштабній структурі

Спочатку зображення з шумом піддається вейвлет-розкладу, що розділяє його на дві субсмуги: низькочастотну La , яка містить базову інформацію, та високочастотну Ha , яка містить деталі. Далі, застосовується аналогічне перетворення для низькочастотної субсмуги. В результаті утворюються дві нові субсмуги: низько-високочастотну LHa та низько-низькочастотну LLa . Потім, низько-низькочастотний сигнал обробляється білатеральним фільтром B для отримання нового низькочастотного сигналу, який перетворюється у нову субсмугу. Для низько-високочастотного сигналу застосовується вейвлет-трешолдинг W , і результат перетворюється в нову субсмугу LHs . Оброблені сигнали об'єднуються за допомогою зворотного вейвлет-перетворення в нову низькочастотну субсмугу LS . До високочастотної субсмуги застосовується вейвлет-трешолдинг, після чого сигнал перетворюється у нову субсмугу HS . Останні сигнали проходять через білатеральний фільтр і комбінуються, щоб отримати фінальне відновлене зображення.

Процес фільтрації базується на вейвлет-розкладі, при якому сигнал розбивається на частотні підсмуги. Білатеральний фільтр застосовується до апроксимаційних підсмуг, що дозволяє ефективно усувати низькочастотні компоненти шуму. У поєднанні з вейвлет-трешолдингом у деталях підсмуг, цей підхід дозволяє успішно видаляти певні шуми.

В рамках процесу фільтрації сигнал розбивається на частотні підсмуги за допомогою вейвлет-розкладання. Оскільки сигнал підлягає відновленню, білатеральний фільтр застосовується до апроксимаційних підсмуг. На відміну від традиційної одношарової білатеральної фільтрації, кратномасштабна білатеральна фільтрація забезпечує можливість усунення низькочастотних компонентів шуму. Фільтрація здійснюється в апроксимаційних підсмугах і, за потреби, може поєднуватися з вейвлет-трешолдингом у підсмугах із деталями, що дозволяє ефективно виявляти та видаляти певні компоненти шуму.

2.3 Експериментальні дослідження розробленого методу

Для оцінки ефективності запропонованої структури проведено моделювання. Для кількісного аналізу були створені зображення з шумом шляхом додавання білого гаусівського шуму з різними стандартними відхиленнями. Потім ці зашумлені зображення оброблялися для зменшення шуму за допомогою кількох алгоритмів. Оцінка результатів проводилася за допомогою показника PSNR. Для візуального порівняння також використовувалися реальні зображення з шумом.

Для кожного тестового зображення було створено три варіанти з різними рівнями шуму, додавши білий гаусівський шум з відхиленнями 10, 20 та 30. Обробка для усунення шуму проводилася чотирма методами. Один з них передбачав використання Bayes Shrink і алгоритму порогової обробки

вейвлет-коефіцієнтів [4]. Було застосовано п'ятирівневе розкладання з оцінкою дисперсії шуму за медіанною методикою [5]. Другий метод використовував білатеральний фільтр [3]. Згідно з результатами експериментів, були вибрані параметри для білатерального фільтра: $\sigma_d = 1,8$, $\sigma_r = 2 \times \sigma_n$, і розмір блоку 11×11 . Третій метод полягає у послідовному застосуванні Bayes Shrink [4] і білатерального фільтра [13]. Цей метод був включений для того, щоб побачити спільну дію Bayes Shrink [24] і білатерального фільтра [3] і порівняти його з запропонованим методом. Четвертий метод є запропонованим методом. Для запропонованого методу, для одно-рівневого розкладання були використані DB8 фільтри в Matlab.

Для частини білатеральної фільтрації запропонованого методу параметри встановлюються таким чином: $\sigma_d = 1,8$, розмір блоку 11×11 , і $\sigma_r = 1,0 \times \sigma_n$ на кожному рівні. У випадку оригінального білатерального фільтру, $\sigma_r = 2 \times \sigma_n$ був кращим вибором. Однак для запропонованого методу це призводить до зниження середніх значень PSNR. Причиною є подвійне використання білатерального фільтра в цьому методі.

Коли значення σ_r велике, текстура в зображенні була згладжена, що проводило до низького значення PSNR. Після декількох експериментів, $\sigma_r = 1,0 \times \sigma_n$ виявилась кращою з точки зору PSNR значень. Для частини запропонованого методу, де застосовується вейвлет-порогове оброблення, використовувався метод Bayes Shrink [4], а дисперсія шуму оцінювалась за медіаною. Для уникнення крайових ефектів зображення були дзеркально розширені перед фільтрацією, а потім обрізані до початкового розміру. Значення PSNR для різних методів обробки представлені в таблиці 2.1. Візуальне порівняння результатів за критерієм PSNR можна побачити на рисунку 2.2.

Таблиця 2.1 – Порівняння різних методів за показником PSNR при різних рівнях шуму.

Зображення	σ	Bayes Shrink [1]	Білатеральний фільтр [3]	[83] після [124]	OWT [2]	Запропонований метод
Barbara 512×512	10	31.25	31.37	30.92	32.18	31.79
	20	27.32	27.02	27.16	27.98	27.74
	30	25.34	24.69	25.23	25.83	25.61
Boats 512×512	10	31.94	32.08	31.93	32.69	32.48
	20	28.69	28.90	28.80	29.52	29.50
	30	27.13	27.50	27.34	27.89	27.77
Goldhill 512×512	10	31.94	32.08	31.93	32.69	32.48
	20	28.69	28.90	28.80	29.52	29.50
	30	27.13	27.50	27.34	27.89	27.77
Peppers 256×256	10	31.49	32.98	31.89	33.18	33.45
	20	27.85	29.07	28.01	29.33	30.20

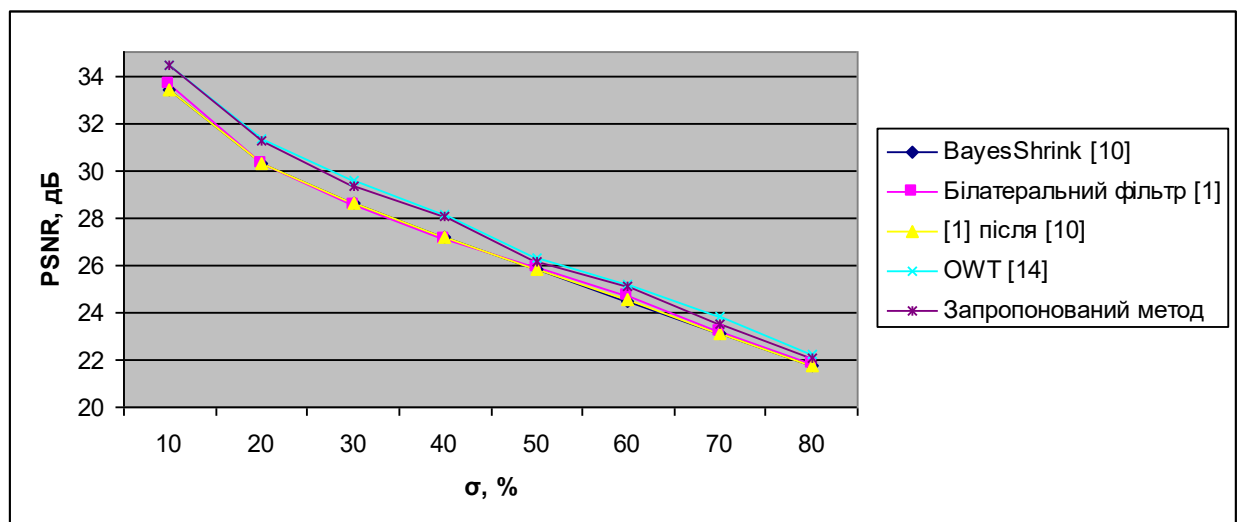


Рисунок 2.2 – Порівняння методів за показником PSNR для зображень з півтонами при різних рівнях шуму

Як показують результати, запропонований метод перевищує ефективність оригінального білатерального фільтра на 0,8 дБ та методу Bayes Shrink у середньому на 1,1 дБ. Послідовне використання методів Bayes Shrink [4] та білатерального фільтра [3] демонструє лише незначну перевагу над самостійним застосуванням Bayes Shrink, але поступається ефективності білатерального фільтра. Отже, можна зробити висновок, що підвищення ефективності запропонованого методу зумовлене саме використанням багаторівневої роздільної здатності у білатеральному фільтрі, а не комбінованим застосуванням Bayes Shrink і білатерального фільтра.

Для досягнення найкращих результатів в умовах реального випадкового шуму, наприклад, фіксованого зразкового шуму, зумовленого високою чутливістю ISO, ефективність запропонованого методу була перевірена на стандартних тестових зображеннях, що містять просторово змінний випадковий шум. Результати оцінки за критерієм PSNR представлені в таблиці 2.2.

Таблиця 2.2 – Порівняння за показником PSNR між різними методами на реальних зображеннях з шумом.

Зображення	Bayes Shrink [2]	Білатеральний фільтр [3]	OWT [6]	Запропонований метод
Barbara 512×512	31.96	32.12	32.26	32.63
Lena 512×512	32.94	33.17	32.87	34.01
Goldhill 512×512	32.06	32.40	32.28	32.76
Boat 512×512	32.05	32.41	32.37	32.93
House 512×512	32.45	34.05	32.39	34.52
Peppers 512×512	31.76	33.42	32.20	33.58
Середнє	32.20	32.93	32.40	33.41

Проаналізувавши отримані результати, можна стверджувати, що цей метод є ефективним для зменшення реального шуму. Водночас, оскільки шум має просторову варіативність, автоматичні оцінки параметрів не можуть бути застосовані в даному випадку.

Однак, виходячи з результатів експерименту, необхідно зрозуміти, що σ_r впливає на продуктивність найбільше, і значення σ_d повинно бути невеликим.

Порівняння за критерієм PSNR відображає лише частину інформації, оскільки, по-перше, відомо, що PSNR не завжди корелює з візуальною якістю зображення, а, по-друге, білий гаусівський шум не завжди точно відображає реальні умови для зображень. Для більш точної оцінки ефективності алгоритмів шумозниження були проведені експерименти з реальними даними та здійснена візуальна перевірка результатів.

Для півтонових зображень змінний просторовий випадковий шум було додано до стандартного тестового зображення "Lena" розміром 512x512 пікселів. Результати порівняння запропонованого методу, білатерального фільтра [13] та методу Bayes Shrink [17] наведені на рисунку 2.3.



а)



б)



в)



г)

Рисунок 2.3 – Візуальне порівняння різних методів при змінному просторовому шумі:

а) Зображення з $\text{PSNR} = 30.02$ дБ, б) Bayes Shrink [24], в) Білатеральний фільтр [3] з $\sigma_d = 1.8$, $\sigma_r = 5\sigma_n$, г) Запропонований метод з однорівневим розкладанням. Парметри $\sigma_d = 1.8$, $\sigma_r = 3\sigma_n$ на кожному рівні.

Для кольорових зображень важливим аспектом є вибір колірного простору для фільтрації. Простір RGB може забезпечити високі значення PSNR, але для зорово комфортного результату краще використовувати перцептивно рівномірний колірний простір CIE-LAB*. Оскільки людська зорова система більш чутлива до кольорового шуму, ніж до яскравісного, до каналів A і B доцільно застосовувати сильнішу фільтрацію порівняно з каналом L (яскравістю).

У проведених експериментах порівнювалися результати фільтрації у просторах RGB і CIE-LAB*. Виявлено, що при обробці в каналах RGB залишаються помітні кольорові шуми. У CIE-LAB*, навпаки, сильне згладжування каналу L може призвести до неприродного вигляду («пластикового» ефекту) через втрату деталей.

Для уникнення цього ефекту запропонований метод і стандартний білатеральний фільтр застосовували до кожного каналу окремо в просторі CIE-LAB*. В експериментах варіювали кількість рівнів розкладання для каналів: для яскравісного каналу L використовували один або два рівні, а для кольорових каналів A і B – чотири. Результати показали, що зображення, оброблені із запропонованими параметрами, майже повністю позбавлені кольорового шуму. Проте, збільшення кількості рівнів розкладання для каналу L з одного до двох призводить до сильнішого згладжування, що може бути небажаним для збереження деталей (див. рис. 2.6е).

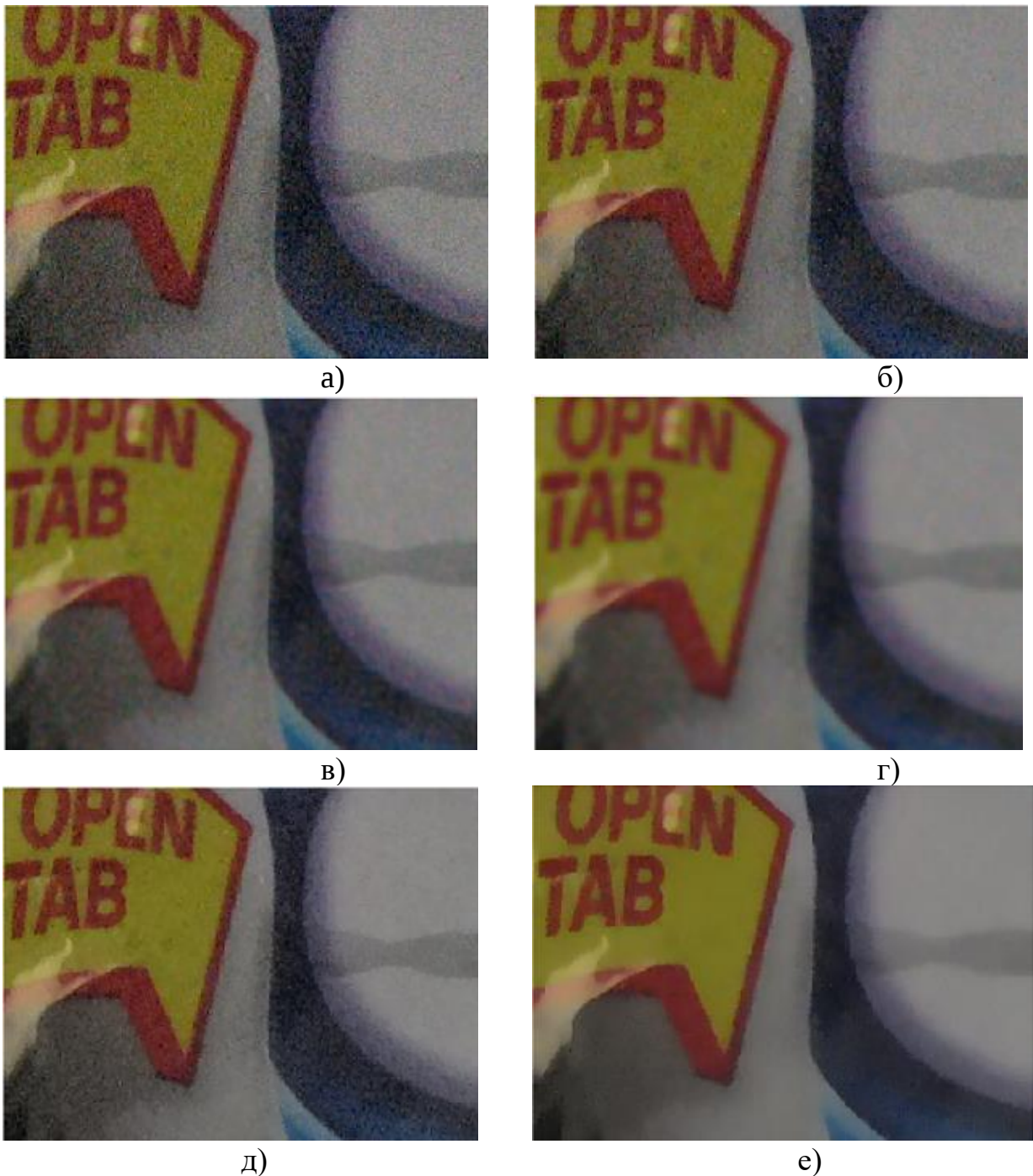


Рисунок 2.4 – Порівняння зображень з реальним шумом:

а) вхідне зображення; б) Білатеральний фільтр [3] з $\sigma_d = 1.8$, $\sigma_r = 3\sigma_n$;
 в) Білатеральний фільтр з $\sigma_d = 1.8$, $\sigma_r = 20\sigma_n$; г) Білатеральний фільтр з
 $\sigma_d = 5$, $\sigma_r = 20\sigma_n$; д) запропонований метод з рівнями декомпозиції (1,4,4)
 для каналів (L, a, b) відповідно, е) запропонований метод з рівнями
 декомпозиції (2,4,4). Параметри $\sigma_d = 1.8$, $\sigma_r = 3\sigma_n$ на кожному рівні.

На рисунках 2.5 і 2.6 продемонстровані візуальні результати очищення реальних зображень за допомогою трьох методів: BLS-GSM [7], білатерального фільтра [8] і запропонованого підходу. Ці результати дозволяють оцінити візуальну якість зображень після фільтрації та порівняти ефективність різних алгоритмів.



а)



б)



в)



г)

Рисунок 2.5 – Порівняння зображень з реальним шумом:

а) оригінальне зображення; б) BLS-GSM [7]; в) білатеральний фільтр [3]; г) запропонований метод. Для білатерального фільтру $\sigma_d = 1.8$, $\sigma_r = 10\sigma_n$ і розмір вікна 11×11 . Для запропонованого методу $\sigma_d = 1.8$,

$\sigma_r = 3\sigma_n$ і розмір вікна 11×11 , кількість

рівнів розкладання (1,4,4) для каналів (L, a, b) відповідно.

Метод BLS-GSM загалом вважається одним із найкращих для очищення від шуму, якщо оцінювати за показником PSNR. Однак, у випадку реальних даних, запропонований метод демонструє візуально більш приємні результати.

На рисунку 2.5 показано, що ні метод BLS-GSM, ні білатеральний фільтр не змогли повністю усунути колірний шум. Натомість запропонований метод ефективніше справляється із цією задачею, забезпечуючи більш природний вигляд зображення.

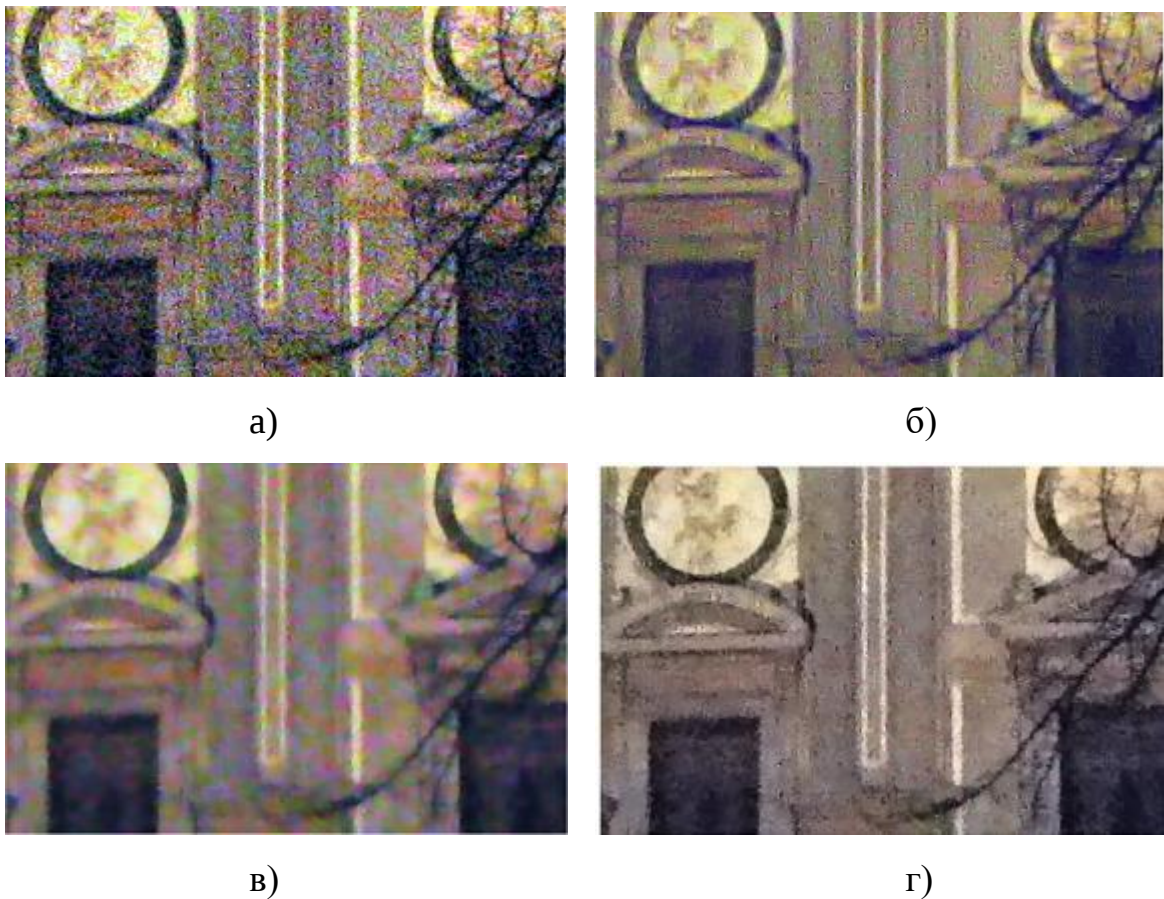


Рисунок 2.6 – Порівняння зображень з реальним шумом:

а) оригінальне зображення; б) BLS-GSM [7]; в) білатеральний фільтр [3]; г) запропонований метод. Для білатерального фільтру $\sigma_d = 1.8$, $\sigma_r = 10\sigma_n$ і розмір вікна 11×11 . Для запропонованого методу $\sigma_d = 1.8$, $\sigma_r = 3\sigma_n$ і розмір вікна 11×11 , кількість

На рисунку 2.6 видно, що метод BLS-GSM не повністю усунув шум, тоді як білатеральний фільтр хоч і демонструє менш шумні результати, але схильний до надмірного згладжування, що призводить до втрати деталей. У порівнянні з цими двома методами, запропонований підхід забезпечує найкращий баланс між видаленням шуму та збереженням текстур, що робить його візуально найбільш прийнятним серед трьох методів.

Запропонований метод демонструє переваги у збереженні деталей і придушенні шуму порівняно з іншими підходами. Водночас видно, що метод BLS-GSM схильний до незначного згладжування текстур, тоді як білатеральний фільтр іноді залишає помітні залишки шуму. Таким чином, запропонований метод поєднує переваги придушення шуму та збереження структури зображення, що підтверджується візуальними порівняннями.

У всіх експериментах з реальними зображеннями значення були оцінені на основі даних, при цьому для запропонованого методу використовувалися однакові значення параметрів. Тобто, після визначення параметрів не було потреби їх змінювати для інших зображень.

Для оцінки якості відновлення кольорових зображень використовується евклідова відстань у перцептивно рівномірному просторі CIE-LAB*. На відміну від PSNR, цей підхід дозволяє враховувати сприйнятливість людського зору до кольорових змін. Зокрема, S-CIELAB метрика, яка базується на CIE-LAB* просторі, є одним із найпоширеніших перцептивних показників. Вона дозволяє оцінити різницю між оригінальним і відновленим зображенням, виражаючи її в одиницях ΔE , де одна одиниця відповідає мінімальній помітній різниці у кольорі.

У цьому експерименті для порівняння продуктивності було взято чотири алгоритми:

- Запропонований метод.
- Оригінальний білатеральний фільтр [3].
- Два вейвлет-алгоритми: Bayes Shrink [4] та SURE Shrink [5].

До стандартних кольорових зображень ("Baboon", "Peppers", "Boat", "Goldhill") було додано білий гаусівський шум з дисперсією 25. Після цього кожен з алгоритмів застосовувався окремо до кожного з трьох кольірних каналів. Продуктивність оцінювалася шляхом вимірювання евклідової відстані між відновленими та оригінальними зображеннями у просторі CIE-LAB*.

Результати експерименту наведено в таблиці 2.3, яка демонструє порівняльну якість алгоритмів. Чим нижче значення ΔE , тим кращий результат відновлення зображення.

Таблиця 2.3 – Порівняння різних методів за допомогою евклідової відстані в просторі CIELAB* [8].

Зображення	BayesShrink	SURE Shrink	Білатеральний фільтр	Запропонований метод
Baboon 512×512	11.9730	13.2110	7.9895	7.6034
Peppers 512×512	8.4389	9.1117	5.1905	2.6680
Boat 512×512	7.4325	7.8173	6.7754	3.5880
Goldhill 512×512	8.6210	9.4387	7.608	5.1807

З таблиці 2.3 видно, що запропонований метод демонструє найменшу кольорову різницю, що вказує на його високу ефективність у збереженні кольору зображення. Це означає, що метод найбільше відповідає вимогам зорової системи людини, оскільки зменшується перцептивна різниця між відновленим і оригінальним зображенням. Такий результат підвищує якість візуального сприйняття відновлених зображень і робить метод більш підходящим для реальних застосувань, де важливі не лише технічні показники (як PSNR), а й візуальна приємність результатів.

Отримані після фільтрації зображення були ущільнені за допомогою алгоритму JPEG з різним показником якості $q = 10 - 90$ з кроком 5.

Графік залежності показника PSNR від q наведений на рисунку 2.7.

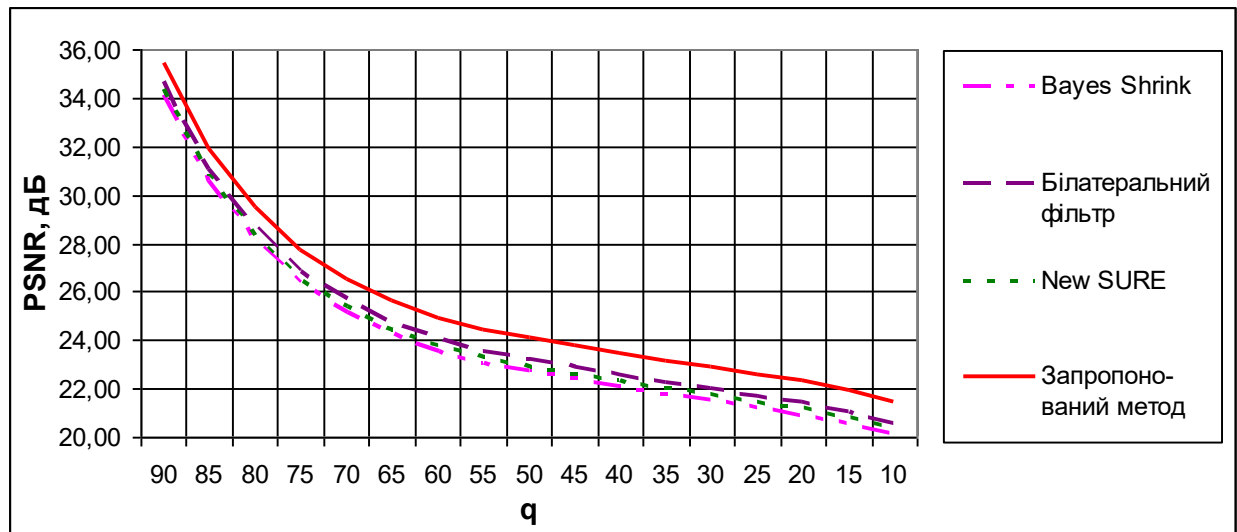


Рисунок 2.7 – Залежність показника PSNR від q для зображень з бази USC-SIPI Image Database

Аналізуючи графік, можна зробити висновок про ефективність запропонованого методу, який демонструє кращі значення PSNR, ніж білатеральний фільтр, метод Bayes Shrink та метод New SURE. Необхідно відмітити, що критичне значення показника PSNR = 25 дБ досягається для запропонованого методу при $q = 60$, а для методів Bayes Shrink, білатерального фільтру та New SURE при $q \approx 65 \dots 70$.

Кожне тестове зображення ущільнювалось за алгоритмом JPEG з максимальним коефіцієнтом ущільнення, але при збереженні показника PSNR > 25 дБ. Для запропонованого методу $q = 60$, для методу Bayes Shrink – $q = 70$, для білатерального фільтру та методу New SURE – $q = 65$.

Результати порівняння за розміром зображень після ущільнення між різними методами при збереженні показника PSNR > 25 дБ наведені в таблиці 2.4.

Таблиця 2.4 – Порівняння за розміром зображень між різними методами при показнику PSNR > 25 дБ.

Зображення	Розмір, кБ					
	Оригінальне	Без фільтрації	Bayes Shrink [127]	Білатеральний фільтр [89]	New SURE [129]	Запропонований метод
Barbara 512×512	257	62	32	29	31	23
Lena 512×512	257	61	31	28	30	23
Goldhill 512×512	257	67	34	32	33	27
Boat 512×512	257	69	38	35	38	29
House 512×512	256	68	37	35	36	30
Peppers 512×512	257	66	35	32	34	28

Для кожного тестового зображення був розрахований коефіцієнт ущільнення за формулою:

$$k = \frac{S_o}{S_c}, \quad (2.14)$$

де k – коефіцієнт ущільнення;

S_o – об'єм вихідних даних;

S_c – об’єм ущільнених даних.

Результати порівняння за коефіцієнтом ущільнення між різними методами при показнику PSNR > 25 дБ наведені в таблиці 2.5.

Таблиця 2.5 – Порівняння за коефіцієнтом ущільнення між різними методами при показнику PSNR > 25 дБ.

Зображення	Коефіцієнт ущільнення					
	Оригінальне	Без фільтрації	Bayes Shrink [127]	Білатеральний фільтр [89]	New SURE [129]	Запропонований метод
Barbara 512×512	1	4,15	8,03	8,86	8,29	11,17
Lena 512×512	1	4,21	8,29	9,18	8,57	11,17
Goldhill 512×512	1	3,84	7,56	8,03	7,79	9,52
Boat 512×512	1	3,72	6,76	7,34	6,76	8,86
House 512×512	1	3,76	6,92	7,31	7,11	8,53
Peppers 512×512	1	3,89	7,34	8,03	7,56	9,18
Середнє	1	3,93	7,48	8,13	7,68	9,74

Аналіз результатів дозволяє зробити висновок, що ефективна фільтрація зображення від шуму перед ущільненням значно покращує коефіцієнт ущільнення. Використання розробленого методу фільтрації перед етапом ущільнення за алгоритмом JPEG дозволило збільшити коефіцієнт ущільнення в середньому на 248% порівняно з ущільненням без фільтрації, а також на 20–30% у порівнянні з іншими методами фільтрації, при збереженні показника якості PSNR > 25 дБ.

2.4 Розроблення узагальненої методики дослідження модуля фільтрації цифрових зображень

Методика дослідження модуля фільтрації цифрових зображень складається з наступних етапів:

1. Визначення типу вхідних зображень (півтонове або кольорове).
2. Фільтрація отриманих зображень.
3. Стиснення кодеком з втратами.
4. Оцінювання коефіцієнта стиснення.
5. Оцінювання якості стиснення зображень.
7. Оцінювання якості отриманих зображень.

Для фільтрації отриманої відеопослідовності:

- а) виконується розбиття відеопослідовності на кадри;
- б) до кожного кадру:
 - у випадку 8-ми бітної півтонової відеопослідовності застосовується кратномасштабний білатеральний фільтр з відповідними налаштуваннями параметрів;
 - у випадку 24-х бітної повнокольорової відеопослідовності:
 - виконується переведення зображення у колірний простір CIE-LAB;
 - виконується фільтрація зображення з використанням відповідного фільтру.
 - виконується зворотне перетворення у колірний простір RGB.

в) виконується компонування відеопослідовності з оброблених кадрів.

Для ущільнення відеопослідовності використовуються наступні кодеки:

- покадрові (MJPEG);

- міжкадрові (H.264, MPEG-4 (XVID)).

Параметри стиснення:

- «висока якість»;
- «середня якість».

Коефіцієнт стиснення оцінюється за формулою (2.14):

Якість отриманої після стиснення відеопослідовності оцінюється за суб'єктивними критеріями на основі експертних оцінок та за кількісним показником – BIQI.

Для суб'єктивного оцінювання якості застосовувалося програмне забезпечення MSU Perceptual Video Quality Tool та метод оцінювання SCACJ (Stimulus Comparison Adjectival Categorical Judgement), описаний у стандарті ITU-R BT.500-11 [55]. Відеоматеріали демонструвалися парами одночасно (див. рисунок 2.8), після чого експерт мав оцінити їх якість, використовуючи порівняльну шкалу (див. рисунок 2.9).

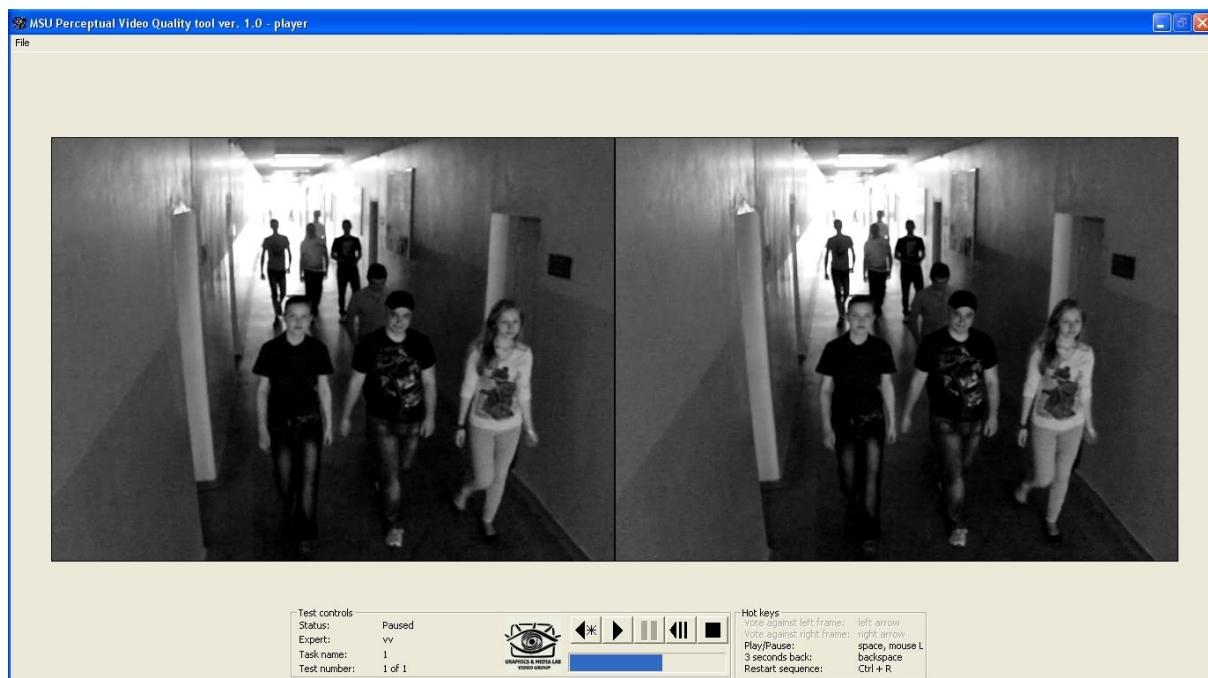


Рисунок 2.8 – Оцінювання якості відео за методом SCACJ

Give your mark! (SCACJ method)

Please, choose your opinion about the quality of the LEFT picture compared to the quality of the RIGHT picture (for example, choosing -2 or -3 means that the LEFT picture is slightly worse than the RIGHT one).

☐ Much worse Worse Slightly worse The same Slightly better Better Much better ☐

-3 -2 -1 0 1 2 3

Circles symbolize your opinion on left and right video correspondingly. Red circle means that video is bad, and green means that video is good.

Your choice: 0

Рисунок 2.9 – Шкала порівняння якості для оцінювання за методом SCACJ

Результати оцінювання зберігаються у файлі *.csv. За результатами оцінювання всіх експертів обчислюється середня суб'єктивна оцінка MOS:

$$MOS_k = \sum_{i=1}^{e_num} \frac{mark_{i,k}}{e_num} \quad (2.15)$$

де k – номер відеопослідовності, для якої обчислюється оцінка;

$mark_{i,k}$ – оцінка, поставлена i -м експертом k -й відеопослідовності;

e_num – загальна кількість експертів.

Для кількісного оцінювання якості відеозображень використовується показник BIQI, який враховує 5 типів спотворень зображення: спотворення, які вносить алгоритм ущільнення JPEG (JPEG); спотворення, які вносить алгоритм ущільнення JPEG2000 (JP2K); шум (NOISE); розмивання зображення (BLUR); спотворення під час передання – fast fading (FF) [26].

На рисунку 2.10 наведено процес оцінювання за показником BIQI.

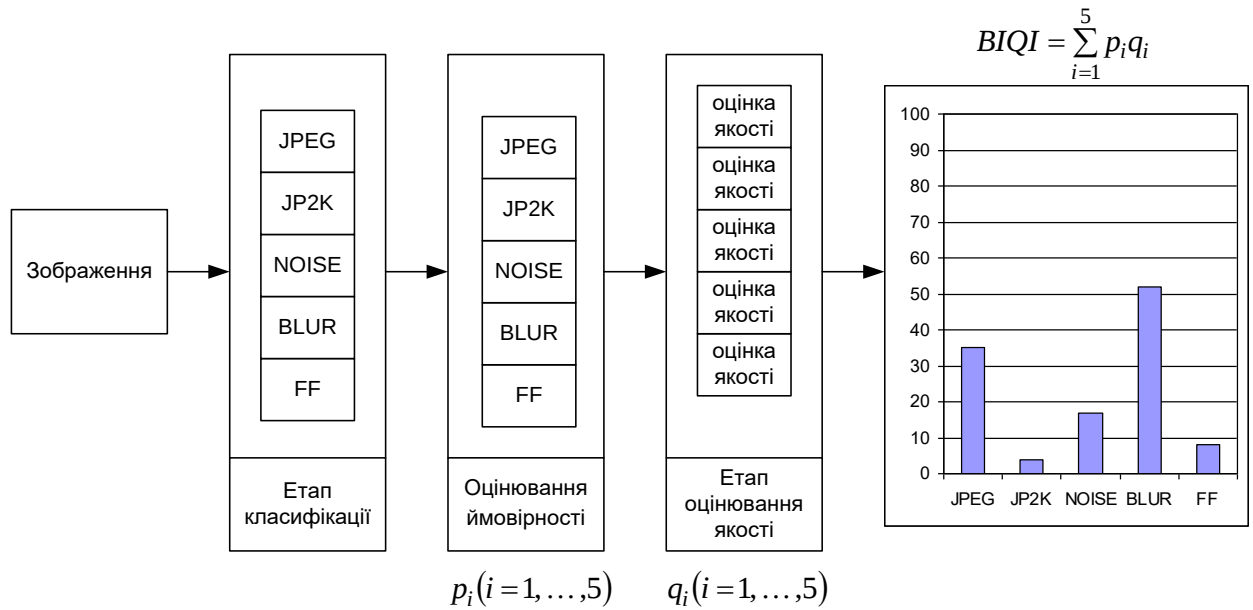


Рисунок 2.10 – Процес оцінювання за показником BIQI

Процес оцінювання складається з наступних етапів:

- визначення типів спотворення;
- класифікація спотворень;
- оцінювання ймовірності кожного спотворення $p_i \{i = 1, \dots, 5\}$;
- оцінювання якості за кожним типом спотворення $q_i \{i = 1, \dots, 5\}$;
- додавання всіх оцінок за формулою:

$$BIQI = \sum_{i=1}^5 p_i q_i, \quad (2.16)$$

де p_i – оцінка ймовірності спотворення;

q_i – оцінка якості;

Значення показника якості BIQI знаходяться в межах від 0 до 100 (0 – найкраща якість, 100 – найгірша якість). Чим менше значення показника BIQI, тим краща якість зображення. Для визначення показника BIQI використовується програмне забезпечення «BIQI Software release» [26].

Для суб'єктивної оцінки якості використовується програмне забезпечення MSU Perceptual Video Quality tool та метод оцінювання SAMVIQ (Subjective Assessment Method for Video Quality evaluation) [25]. Під час тесту експерт може програвати будь-яку послідовність з тестового набору і давати їй свою оцінку (рисунок 2.11).

The screenshot shows a software window titled "Give your mark! (SAMVIQ method)". Inside, it prompts the user to "Please, vote on the video that you have just watched (reference video is not evaluated)". A vertical slider scale is displayed with numerical values from 0 to 100 in increments of 20. To the right of the scale are qualitative labels: "Excellent" at 100, "Good" at 80, "Fair" at 60, "Poor" at 40, and "Bad" at 0. A slider handle is positioned at the 55 mark. Below the scale, instructions state: "Now if you have ranked all the videos you can finish this task. If there are still some sequences left or you want to change your opinion on some sequence, choose a sequence by pressing one of the buttons below." Three buttons are shown: "Change mark", "Watch video", and "Finish task". At the bottom, three boxes represent different video sequences: "Reference" (labeled "Reference"), "[A]" (labeled "new mark 55"), and "B" (labeled "mark not set").

Рисунок 2.11 – Шкала порівняння якості для оцінювання за методом SAMVIQ

Результати оцінювання зберігаються у файлі *.csv. За результатами оцінювання всіх експертів обчислюється середня суб'єктивна оцінка MOS за формулою (2.15). Для кількісного оцінювання якості відеозображень використовується показник VIQI.

2.5 Висновки до другого розділу

Існує велика кількість різних методів фільтрації, і кожен з них має свої переваги та недоліки. Вибір методу залежить від специфіки завдання та вимог до результату. Основною метою методів фільтрації є ефективне усунення шумових компонент зображення, при цьому зберігаючи максимальну кількість корисної інформації та мінімізуючи спотворення.

Білатеральний фільтр є одним з популярних методів, що поєднує просторову фільтрацію і фільтрацію на основі інтенсивності пікселів. Він має перевагу у збереженні контурів та деталей зображення, що робить його корисним для фільтрації шумів при збереженні високої якості зображення. Однак, його ефективність обмежена при роботі з сильно зашумленими зображеннями або складними шумовими структурами.

Запропонований процес фільтрації, що використовує кратномасштабний аналіз та білатеральну фільтрацію, дозволяє покращити результат, знижуючи рівень шуму при збереженні деталей зображення. Кратномасштабний аналіз дозволяє обробляти зображення на різних рівнях роздільної здатності, що дає змогу ефективніше обробляти шум на різних масштабах. Це дозволяє значно покращити якість відновлених зображень у порівнянні з використанням традиційних методів фільтрації.

Таким чином, поєднання кратномасштабного аналізу та білатеральної фільтрації дає значні переваги в обробці зображень, зокрема у випадках, коли потрібно усунути шум при збереженні деталей та контурів.

Запропонований метод фільтрації, що поєднує кратномасштабний аналіз та білатеральну фільтрацію, є доцільним для розробки модуля фільтрації цифрових зображень у системах автоматизації, зокрема для відділів відеоаналітики підприємства. Такий метод може бути ефективно використаний в автоматичних системах обробки відео та зображень для виконання завдань, пов'язаних з очищенням відеопотоків від шуму,

поліпшенням якості зображень та забезпеченням точності візуальних аналізів.

У таких системах, де важливо отримати чіткі зображення з високою контрастністю та деталями, особливо при наявності різних типів шуму, білатеральний фільтр з кратномасштабним аналізом може суттєво покращити якість обробки відео. Наприклад, для відеоаналітики, що здійснює автоматичне розпізнавання об'єктів, моніторинг безпеки або контроль виробничих процесів, зниження рівня шуму при збереженні ключових деталей зображення є критично важливим для досягнення високої точності в алгоритмах обробки.

Кратномасштабний аналіз дозволяє адаптувати фільтрацію до різних рівнів деталізації, що дає змогу ефективно працювати з відео з різною роздільною здатністю та умовами освітлення. Білатеральна фільтрація допомагає зберегти контури та текстури, що є важливими для точного виявлення та аналізу об'єктів на зображенні.

Використання цього методу для фільтрації зображень в системах автоматизації відеоаналітики дозволяє досягти значних переваг, таких як зменшення помилок, покращення якості відеоаналізу та скорочення часу обробки, що особливо важливо для реальних часів та вимог до високої продуктивності системи.

3 РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Структура процесу фільтрації зображення

Процес роботи системи відеоспостереження з фільтрацією зображень та їх збереженням у сховищі даних для подальшого використання можна описати наступним чином:

1. Камери встановлюються для зйомки та передачі відеоданих. Вони є джерелом зображень, які передаються в систему.
2. Отримані з камер дані проходять попередню обробку. Цей етап включає корекцію кадрів, покращення якості зображень, а також підготовку їх для подальшої обробки.
3. Оброблені кадри передаються на модуль фільтрації, який усуває шуми, артефакти та інші недоліки, що впливають на якість відео. Після цього фільтровані дані готові до використання.
4. Відфільтровані зображення зберігаються в сховищі даних. Це дозволяє забезпечити доступ до записів у майбутньому.
5. Через інтерфейс моніторингу надається доступ до записаних даних або даних у реальному часі. Це дозволяє операторам переглядати відео та аналізувати інформацію.
6. Система також передбачає можливість передачі зображень у реальному часі з камер через модуль фільтрації прямо в інтерфейс моніторингу, що забезпечує оперативність доступу до поточної ситуації.

Структура роботи підсистеми відеоспостереження з модулем фільтрації зображень представлена на рисунку 3.1.



Рисунок 3.1 – Структура роботи підсистеми відеоспостереження з модулем фільтрації зображень

Схема фільтрації цифрових зображень представлена на рисунку 3.2.

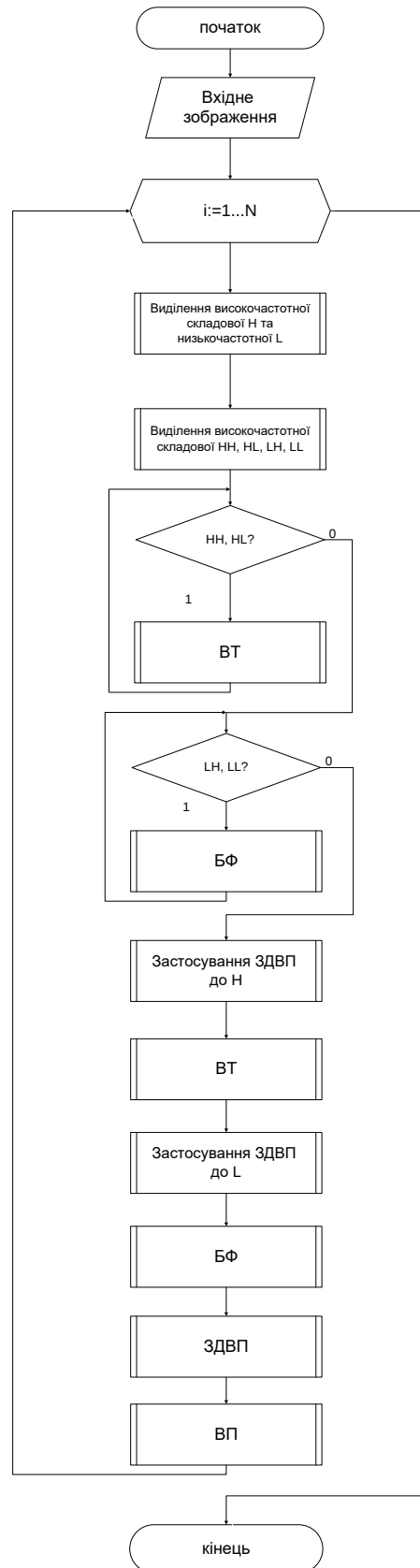


Рисунок 3.2 – Метод фільтрації, що використовує білатеральний фільтр. Блок-схема програми.

Процес обробки зображення за даним алгоритмом можна поділити на такі етапи:

1. Завантаження початкового зображення.
2. Виконання діадного вейвлет-перетворення для розкладу зображення на низькочастотну та високочастотну компоненти.
3. Додаткове розкладання низькочастотної компоненти на дві субсмуги: низько-високочастотну та низько-низькочастотну.
4. Застосування білатерального фільтра до низько-високочастотної субсмуги.
5. Використання вейвлет-трешолдингу для високочастотної компоненти.
6. Проведення зворотного вейвлет-перетворення для відновлення зображення з субсмуг низької та високої частоти.
7. Застосування білатерального фільтра до низько-низькочастотної субсмуги.
8. Застосування вейвлет-трешолдингу до низькочастотної компоненти.
9. Виконання зворотного вейвлет-перетворення для двох субсмуг низької частоти.
10. Застосування білатерального фільтра до результатів.

Отримання фінального покращеного зображення та його збереження.

Процес фільтрації цифрових зображень зі штучним шумом наведений на рисунку 3.3.

Діаграма класів наведена на рисунку 3.4.

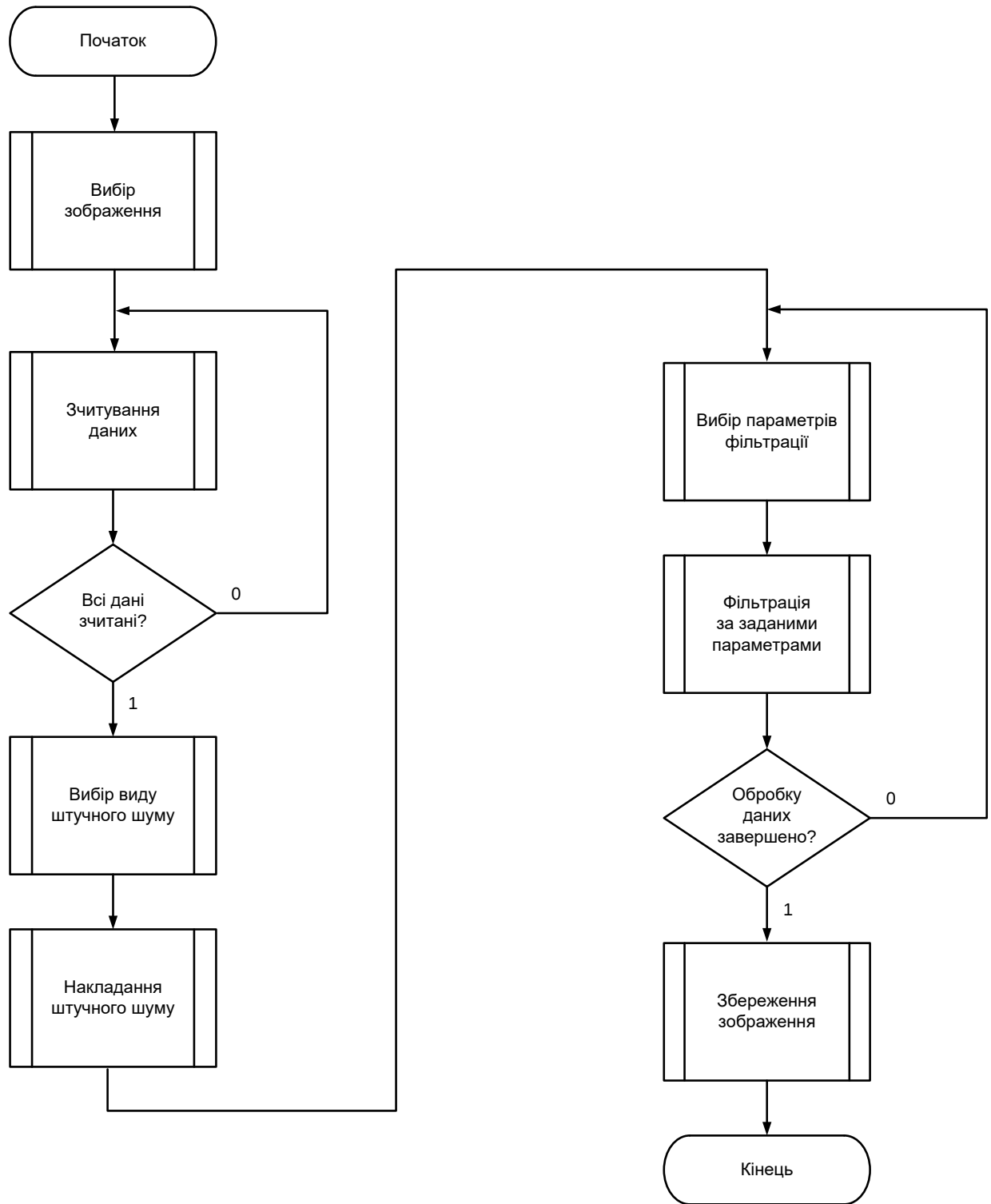


Рисунок 3.3 – Процес фільтрації цифрових зображень зі штучним шумом.

Схема програми.

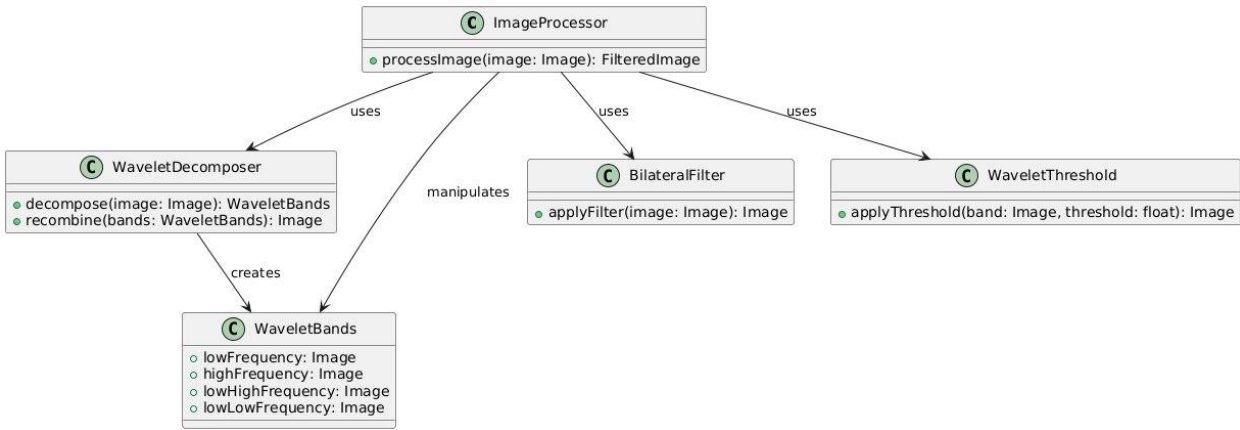


Рисунок 3.4 – Діаграма класів процесів фільтрації цифрових зображень зі штучним шумом

Діаграма класів методики дослідження модуля фільтрації цифрових зображень наведена на рисунку 3.5.

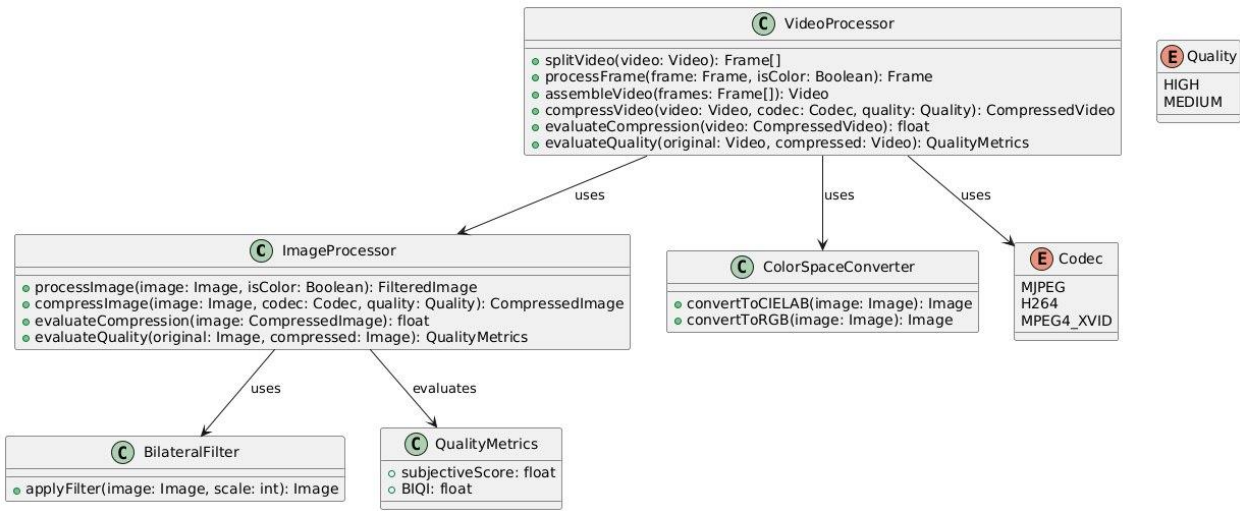


Рисунок 3.5 – Діаграма класів методики дослідження модуля фільтрації цифрових зображень

Процес фільтрації цифрових зображень зі штучним шумом наведений на рисунку 3.6.

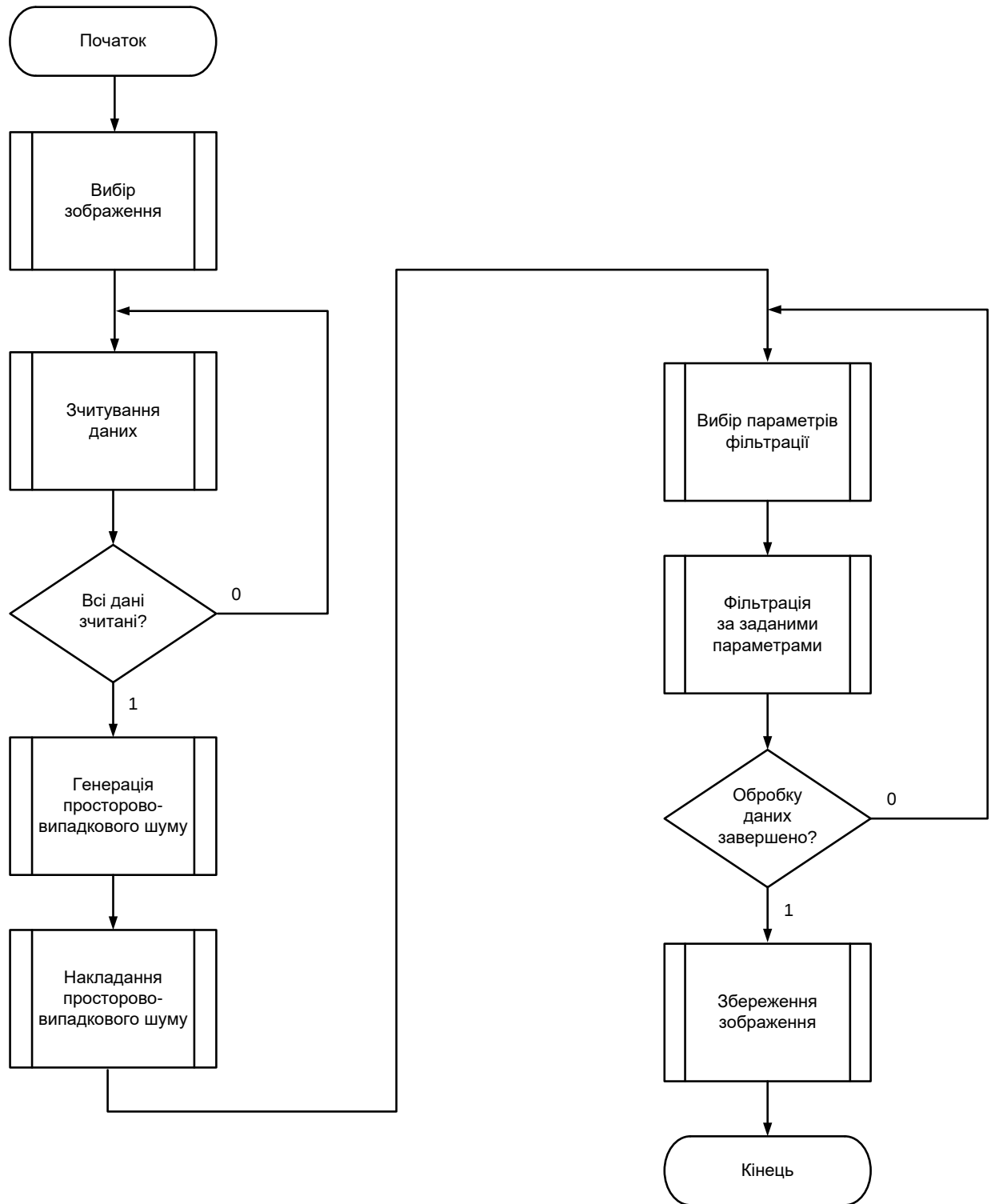


Рисунок 3.6 – Процес фільтрації цифрових зображень з просторово-випадковим шумом. Схема програми.

Use case діаграма наведена на рисунку 3.7.



Рисунок 3.7 – Use case діаграма процесу фільтрації цифрового зображення

3.2 Розробка програмного забезпечення

Програмне забезпечення розроблено на мові C++ в середовищі Microsoft Visual Studio 2022. Для його роботи потрібен ПК класу Pentium IV або вище з операційною системою Windows 2000/XP/Vista/7/8; мінімальний обсяг ОЗП – 512 Мбайт. Розмір інсталяційного файлу – 1,5 Мбайт. Інтерфейс програми українською мовою. Головне вікно програми показано на рисунку 3.8. Програма дозволяє вибирати файли для обробки через навігаційний файловий менеджер, зображений на рисунку 3.9.

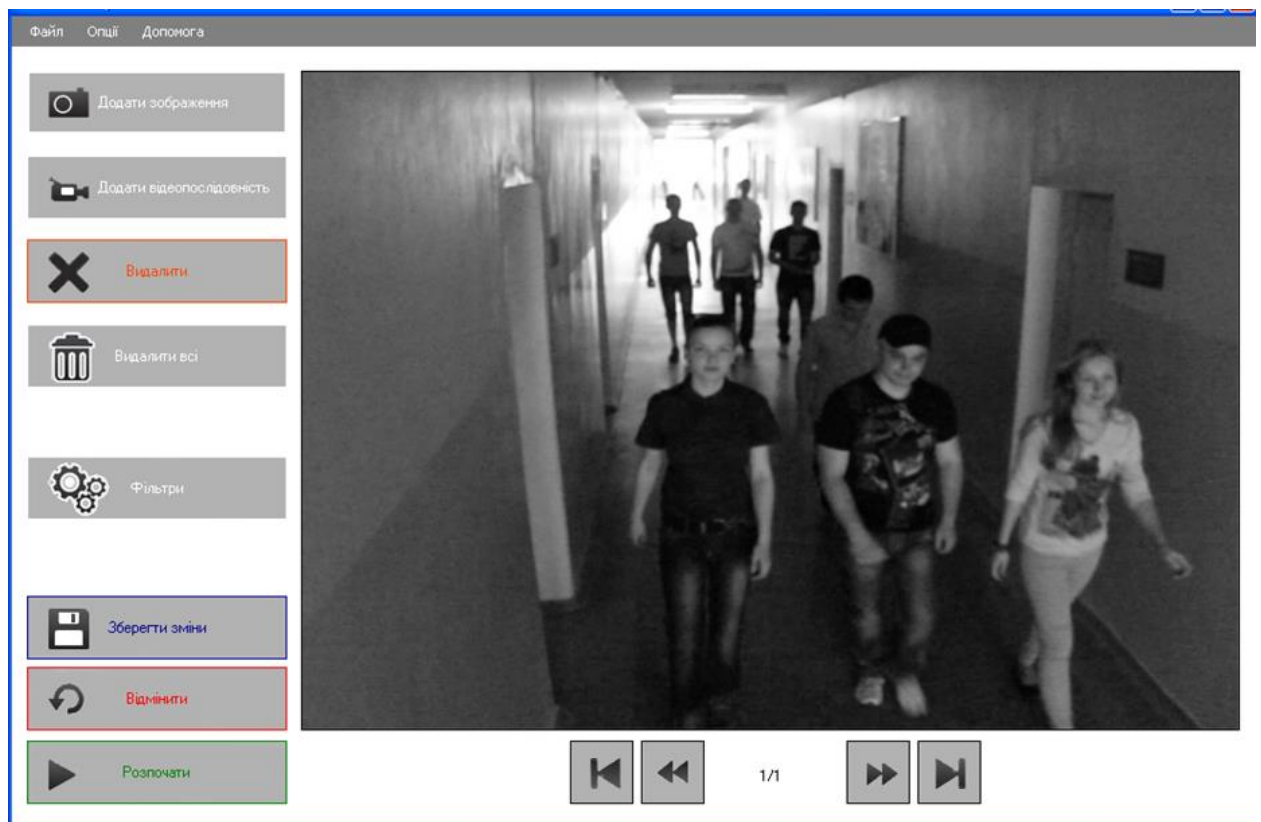


Рисунок 3.8 – Головне вікно програми

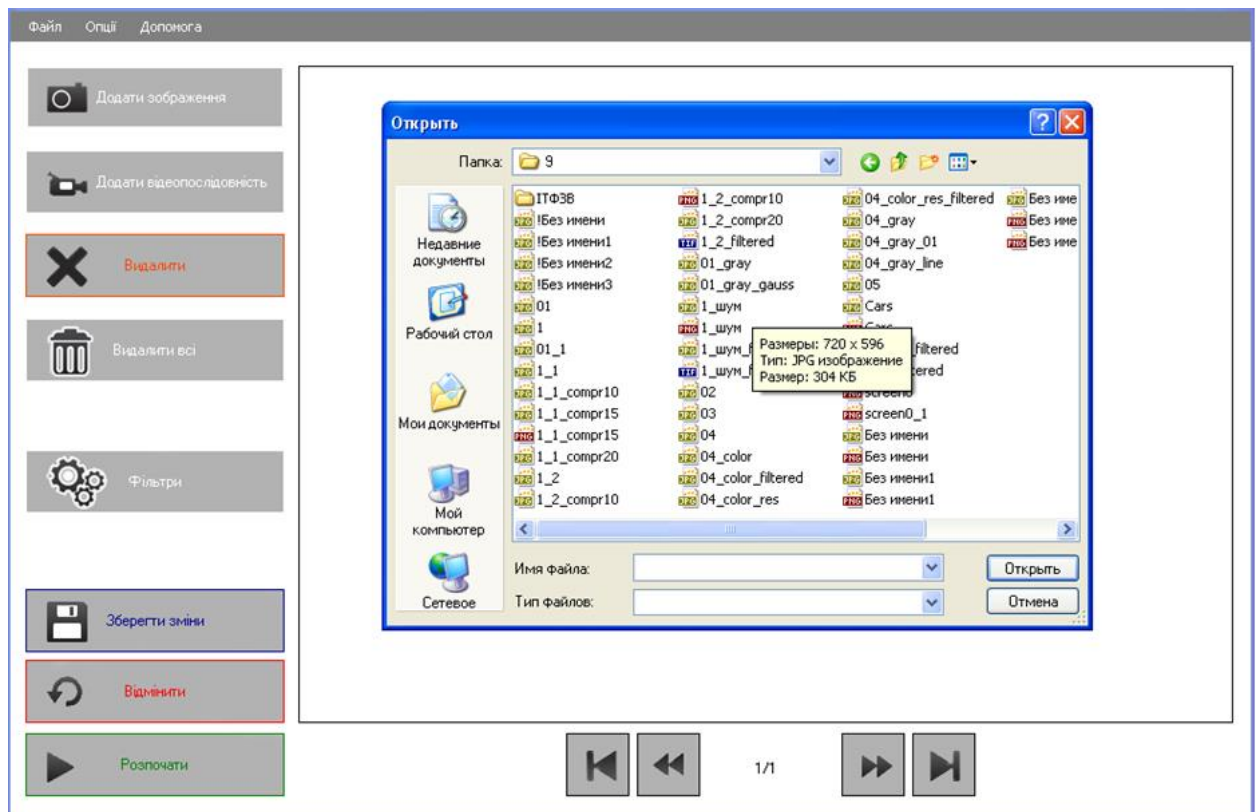


Рисунок 3.9 – Панель вибору файлів для оброблення

Вихідними даними для програми є цифрові зображення у форматах BMP, JPG, JPEG, JPE, TIFF, TIF, PNG. За допомогою чотирьох кнопок на головній формі можна переглядати вибране зображення або послідовність зображень, а також вибирати зображення для обробки.

Конкретні процедури обробки зображень вибираються через кнопку «Фільтри», яка відкриває панель для вибору методу та параметрів фільтрації, що показана на рисунку 3.8.

Для методу фільтрації передбачено можливість встановлення відповідних опцій та параметрів фільтрації. Також реалізовано попередній перегляд зображення після вибору параметрів фільтрації, а також можливість встановлення параметрів за замовчуванням.

Для застосування методу КБФ доступні наступні параметри налаштування (див. рисунок 3.10):

1. Розмір блоку.

2. Параметр контролю зменшення ваги у просторовій області σ_d .
3. Параметр контролю зменшення ваги у просторовій області σ_r .

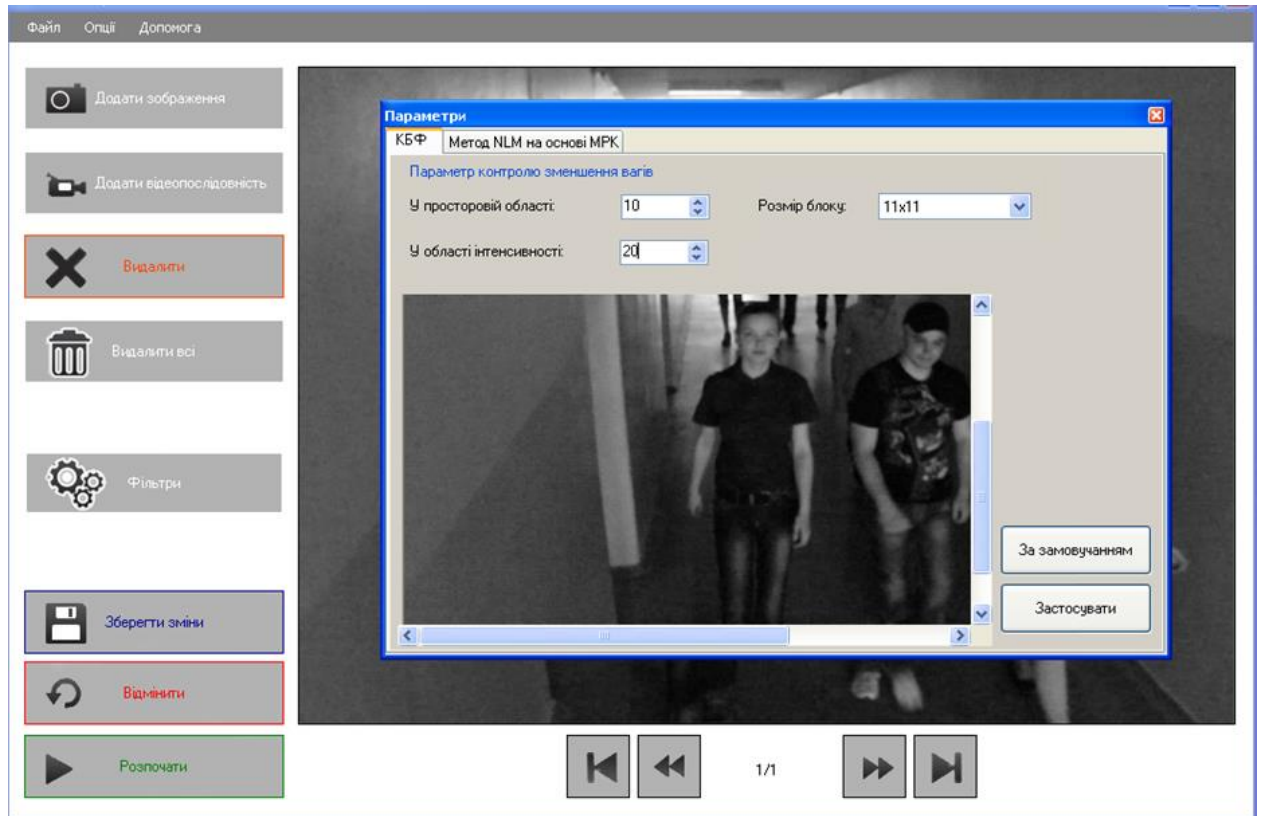


Рисунок 3.10 – Вибір параметрів фільтрації

Лістинг модуля програми наведений в додатку В.

3.3 Висновки до третього розділу

У цьому розділі було розроблено та описано алгоритмічне і програмне забезпечення для модуля фільтрації цифрових зображень, який інтегрується у систему автоматизації відділу відеоаналітики підприємства.

Було створено деталізовану схему роботи системи відеоспостереження з урахуванням етапів обробки, фільтрації та збереження зображень у базі

даних. Це дозволяє забезпечити якісну попередню обробку даних, що впливає на точність подальшої аналітики.

Реалізовано алгоритм фільтрації цифрових зображень, що базується на білатеральному фільтрі, діадному вейвлет-перетворенні та методі трешолдингу. Такий підхід дозволяє ефективно усувати шуми, покращувати якість зображень і зберігати важливу структурну інформацію.

Створено гнучке та функціональне програмне забезпечення на мові C++ в середовищі Microsoft Visual Studio 2022

Структура та функціонал програмного забезпечення дозволяють легко адаптувати його до специфічних завдань підприємства, розширювати функціональні можливості та інтегрувати нові методи обробки.

Розроблений модуль є важливим компонентом системи відеоаналітики, що дозволяє не лише зберігати якісні дані для подальшого аналізу, а й забезпечує моніторинг у реальному часі. Це сприяє підвищенню ефективності роботи відділу відеоаналітики.

Таким чином, у межах даного розділу було розроблено потужний інструмент для фільтрації цифрових зображень, який забезпечує високу якість обробки, зручність у використанні та широкі можливості інтеграції в автоматизовані системи підприємства.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [39].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних
Ринкові переваги (недоліки)					
	Багато аналогів на малому	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
	Ціна продукту значно вища за ціни	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
	Технічні та споживчі властивості продукту значно гірші,	Технічні та споживчі властивості продукту трохи гірші,	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 4.1

Ринкові перспективи					
	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у
1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
2	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	4
2. Ринкові переваги (наявність аналогів)	4	3	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	38	40	38
Середньоарифметична сума балів CB_c	38,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [39].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» становить 38,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [40]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і

при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{na} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю- вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
PSNR для штучного шуму	Дб	29,07	30,2	1,04	0,2
PSNR для реального шуму	Дб	32,93	33,41	1,01	0,3
евклідова відстань в CIEL*A*B* просторі	0	7,608	5,18	1,47	0,15
максимальна роздільна здатність	пікс	2048x2048	2048x2048	1	0,25
рівень шуму	%	5...70	5...70	1	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,04 \cdot 0,2 + 1,01 \cdot 0,3 + 1,47 \cdot 0,15 + 1 \cdot 0,25 + 1 \cdot 0,1 = 1,08.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,08 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [40]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=22$ дні.

$$Z_o = 20000,00 \cdot 10 / 22 = 9090,91 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Менеджер проекту (координатор)	20000,00	909,09	10	9090,91
Інженер-розробник програмного забезпечення	22000,00	1000,00	15	15000,00
Аналітик	22000,00	1000,00	10	10000,00
Інженер-розробник АСУ	17500,00	795,45	15	11931,82
Всього				46022,73

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [40];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000,00 \cdot 1,10 \cdot 1,15 / (22 \cdot 8) = 57,50 \text{ грн.}$$

$$З_{pl} = 57,50 \cdot 12,00 = 690,00 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обладнання для розробки програмного забезпечення	12,00	2	1,10	57,50	690,00
Підготовка робочого місця розробника програмного забезпечення	8,50	3	1,35	70,57	599,83
Інсталяція програмного забезпечення для розпізнавання і фільтрування зображення	6,20	5	1,70	88,86	550,95
Компіляція програмних блоків	8,00	4	1,50	78,41	627,27
Налагодження програмних блоків	4,00	5	1,70	88,86	355,45
Формування бази даних для проведення цифрового експерименту	15,00	4	1,50	78,41	1176,14
Попереднє тренування системи	60,00	3	1,35	70,57	4234,09
Всього					8233,74

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{\text{доо}} = (З_o + З_p) \cdot \frac{H_{\text{доо}}}{100\%}, \quad (4.7)$$

де $H_{\text{доо}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$З_{\text{доо}} = (46022,73 + 8233,74) \cdot 10 / 100\% = 5425,65 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{\text{доо}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.8)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (46022,73 + 8233,74 + 5425,65) \cdot 22 / 100\% = 13130,06 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\epsilon j}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\epsilon j}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 4,0 \cdot 215,00 \cdot 1,1 - 0 \cdot 0 = 946,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір SOFT Ultra Plus	215,00	4,0	0	0	946,00
Папір для записів Parers SOFT Light A5	136,00	5,0	0	0	748,00
Органайзер офісний OFFICE SOFT	198,00	2,0	0	0	435,60
Канцелярське приладдя (набір офісного працівника)	185,00	3,0	0	0	610,50
Картридж для принтера Егіхон EZ2500	860,00	1,0	0	0	946,00
Диск оптичний NewOice CD-RW	22,00	3,0	0	0	72,60
Flesh-пам'ять Kingston 64 GB	623,00	1,0	0	0	685,30
Тека для паперів BOSS BOX	75,00	5,0	0	0	412,50
Інші матеріали	195,00	1,0	0	0	214,50
Всього					5071,00

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.10)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$$K_6 = 1 \cdot 1199,00 \cdot 1,1 = 1318,90 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
SSD диск Kingston SSDNow A400 240GB 2.5» SATAIII 3D TLC (SA400S37/240G)	1	1199,00	1318,90
Відеокарта MSI PCI-Ex GeForce RTX 3060 Ventus 2X 12G OC 12GB GDDR6 (192bit) (1807/15000) (HDMI, 3 x DisplayPort) (RTX 3060 VENTUS 2X 12G OC	1	13199,00	14518,90
Маршрутизатор Asus RT-AX57 (90IG06Z0-MO3C00 / 90IG06Z0-MU2C00)	1	4499,00	4948,90
Всього			20786,70

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.11)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.і}}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 17496,00 \cdot 1 \cdot 1,1 = 19245,60 \text{ грн.}$$

Отримані результати зведемо до таблиці.

Таблиця 4.9 – Витрати на придбання спекустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Сервер Dell R630 8 SFF (2*E5-2640v4, 64GB)	1	17496,00	19245,60
Мультимедіа проектор SAMSUNG PLC-XW300	1	45430,00	49973,00
Всього			69218,60

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{npz} = \sum_{i=1}^k C_{inpe} \cdot C_{npz.i} \cdot K_i, \quad (4.12)$$

де C_{inpe} – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{npz} = 11320,00 \cdot 1 \cdot 1,1 = 12452,00 \text{ грн.}$$

Отримані результати зведемо до таблиці.

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
середовище розробки Visual Studio	1	11320,00	12452,00
Програмне забезпечення аналізу відеозображення	1	9650,00	10615,00
Програмне забезпечення програмування C# та технології ML.NET	1	5100,00	5610,00
Програмне забезпечення програмування .NET Core	1	3200,00	3520,00
Абонентна плата (доступ Internet)	1	360,00	396,00
Всього			32593,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{\bar{o}}}{T_{\bar{o}}} \cdot \frac{t_{вик}}{12}, \quad (4.13)$$

де $Ц_{\bar{o}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\bar{o}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (24800,00 \cdot 1) / (3 \cdot 12) = 688,89 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника програмного забезпечення	24800,00	3	1	688,89
Робоче місце інженера-програміста	8250,00	3	1	229,17
Пристрої передачі даних	6540,00	6	1	90,83
Оргтехніка	7850,00	5	1	130,83
Приміщення лабораторії розробки інформаційних систем	205000,00	20	1	854,17
Електронно-обчислювальний комплекс обробки зображень на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642	75900,00	3	1	2108,33
OS Windows 11	6800,00	3	1	188,89
Прикладний пакет Microsoft Office	6700,00	3	1	186,11
Всього				4477,22

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 10,98$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,45 \cdot 90,0 \cdot 10,98 \cdot 0,95 / 0,97 = 444,69 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника програмного забезпечення	0,45	90,0	444,69
Робоче місце інженера-програміста	0,25	90,0	247,05
Пристрої передачі даних	0,03	90,0	29,65
Оргтехніка	0,65	3,0	21,41
Електронно-обчислювальний комплекс обробки зображень на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642	0,25	85,0	233,33
Мультимедіа проектор SAMSUNG PLC-XW300	0,22	20,0	48,31
Сервер Dell R630 8 SFF (2*E5-2640v4, 64GB)	0,45	90,0	444,69
Маршрутизатор Asus RT-AX57 (90IG06Z0-MO3C00 / 90IG06Z0-MU2C00)	0,03	90,0	29,65
Всього			1498,77

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» відсутні.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\epsilon} = (3_o + 3_p) \cdot \frac{H_{i\epsilon}}{100\%}, \quad (4.15)$$

де $H_{i\epsilon}$ – норма нарахування за статтею «Інші витрати», прийmemo $H_{i\epsilon} = 55\%$.

$$I_{\epsilon} = (46022,73 + 8233,74) \cdot 55 / 100\% = 29841,06 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.16)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (46022,73 + 8233,74) \cdot 100 / 100\% = 54256,47 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{доо}} + Z_n + M + K_{\text{в}} + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_{\text{в}} + B_{\text{нзв}}. \quad (4.17)$$

$$\begin{aligned} B_{\text{заг}} = & 46022,73 + 8233,74 + 5425,65 + 13130,06 + 5071,00 + 20786,70 \\ & + 69218,60 + 32593,00 + 4477,22 + 1498,77 + 0,00 + 0,00 + 29841,06 + 54256,47 \\ = & 290554,99 \text{ грн.} \end{aligned}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,9$.

$$ZB = 290554,99 / 0,9 = 322838,88 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження

результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	700	900	1200	1000

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 6500 осіб;

Π_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 4600,00 грн;

$\pm \Delta \Pi_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo -325,10 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [40]:

$$\Delta \Pi_i = (\pm \Delta \Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (-325,10 \cdot 6500,00 + 4274,90 \cdot 700) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 239375,19 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (-325,10 \cdot 6500,00 + 4274,90 \cdot 1600) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1286794,09 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (-325,10 \cdot 6500,00 + 4274,90 \cdot 2800) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2683352,62 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (-325,10 \cdot 6500,00 + 4274,90 \cdot 3800) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 3847151,39 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.20)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,25$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 239375,19/(1+0,25)^1 + 1286794,09/(1+0,25)^2 + 2683352,62/(1+0,25)^3 + \\ &+ 3847151,39/(1+0,25)^4 = 191500,15 + 823548,21 + 1373876,54 + 1575793,21 = 3964718,11 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ЗВ, \quad (4.21)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 322838,88 грн.

$$PV = k_{инв} \cdot ЗВ = 2 \cdot 322838,88 = 645677,76 \text{ грн.}$$

Абсолютний економічний ефект E_{abc} для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV \quad (4.22)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 3964718,11 грн;
 PV – теперішня вартість початкових інвестицій, 645677,76 грн.

$$E_{abc} = III - PV = 3964718,11 - 645677,76 = 3319040,36 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.23)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 3319040,36 грн;

PV – теперішня вартість початкових інвестицій, 645677,76 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 3319040,36/645677,76)^{1/4} - 1 = 0,57.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\min}$:

$$\tau_{\min} = d + f, \quad (4.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,11$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,15.

$\tau_{\min} = 0,11 + 0,15 = 0,26 < 0,57$ свідчить про те, що внутрішня економічна дохідність інвестицій E_{ϵ} , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_{\epsilon}}, \quad (4.25)$$

де E_{ϵ} – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,57 = 1,74 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства» становить 38,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,08 рази.

Також термін окупності становить 1,74 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства».

ВИСНОВКИ

У результаті проведеної роботи досягнута основна мета — створення модуля фільтрації цифрових зображень, що значно підвищує ефективність обробки даних у системах автоматизації. Усі завдання були успішно виконані.

Аналіз сучасних методів фільтрації зображень показав, що більшість методів орієнтовані на півтонові зображення з додаванням штучного шуму, в той час як методи для обробки реальних кольорових фотографій з природним шумом є або складними, або недостатньо ефективними.

Було описано білатеральний фільтр, визначені його переваги та особливості використання. З одного боку, його проста формалізація та виконання: піксель замінюється на зважене середнє значення сусідів. Білатеральний фільтр є не ітераційним, що дозволяє досягти бажаного результату після одного проходження, і параметри фільтру легко налаштовуються. З іншого боку, цей метод є нелінійним і потребує поліпшень для практичного застосування, що вимагає великих обчислювальних ресурсів. Запропоновано процес фільтрації з використанням кратномасштабного аналізу і білатерального фільтру.

Розроблений метод фільтрації адаптовано для застосування в системах відеоаналітики, що дозволяє ефективно зменшувати шум на кольорових фотографіях із реальним шумом.

Експериментальне дослідження показало, що запропонований кратномасштабний білатеральний фільтр дозволяє підвищити PSNR в середньому на 0,1 – 1,12 дБ для зображень з адитивним гаусівським шумом і на 0,48 – 1,21 дБ для зображень зі змінним просторовим шумом у порівнянні з іншими методами.

Реалізовано алгоритмічне та програмне забезпечення модуля фільтрації, що забезпечує швидку та точну обробку зображень без

необхідності ітерацій, досягаючи високої ефективності та оптимізації обчислювальних ресурсів.

Згідно з дослідженнями, рівень комерційного потенціалу розробки за темою «Розробка модуля фільтрації цифрових зображень для систем автоматизації відділу відеоаналітики підприємства» становить 38,7 бала, що вказує на значну комерційну важливість цієї роботи.

За технічними параметрами науково-технічна розробка переважає існуючі аналоги на 1,08 рази, а термін окупності складає 1,74 року, що менше 3 років, підтверджуючи комерційну привабливість розробки та її потенціал для інвестування.

Отже, завдання виконано успішно, і мета проекту досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Добеши І. Обробка зображень в системах безпеки / І. Добеши. — Харків: ХНУРЕ, 2015. — 320 с.
2. Донохо Д. А., Браун Е. Л. Статистичні методи обробки зображень / Д. А. Донохо, Е. Л. Браун. — Київ: Видавничий дім «КМ Академія», 2008. — 360 с.
3. Pan Y., Wang S. Image denoising based on non-local means and wavelet thresholding // Journal of Visual Communication and Image Representation. — 2013. — Vol. 24, Issue 4. — P. 416–426. — URL: <https://doi.org/10.1016/j.jvci.2013.02.005> (дата звернення: 10.12.2024).
4. Zhang K., Li X. Image denoising based on the non-local means algorithm: A comprehensive review // Journal of Visual Communication and Image Representation. — 2020. — Vol. 75. — URL: <https://doi.org/10.1016/j.jvci.2020.102834> (дата звернення: 10.12.2024).
5. Zhou W., Yang C. Image Processing Techniques in Intelligent Surveillance Systems // IEEE Transactions on Systems, Man, and Cybernetics: Systems. — 2019. — Vol. 49, No. 3. — P. 598–610. — URL: <https://doi.org/10.1109/TSMC.2018.2857516> (дата звернення: 10.12.2024).
6. Коваленко М. В. Цифрова обробка зображень для систем відеоспостереження / М. В. Коваленко. — Київ: Наукова думка, 2011. — 340 с.
7. Сапожніков С. В., Лисенко Т. П. Теорія та практика обробки зображень: підручник / С. В. Сапожніков, Т. П. Лисенко. — Харків: ХНУРЕ, 2018. — 290 с.
8. Баранник О. В. Основи цифрової обробки зображень та їх застосування в системах безпеки / О. В. Баранник. — Львів: Вид-во Львівської політехніки, 2016. — 350 с.

9. Tsaftaris S. A., Kappas M. Image Denoising using Sparse Representations and Non-Local Means // IEEE Transactions on Image Processing. — 2017. — Vol. 26, No. 5. — P. 2357–2369. — URL: <https://doi.org/10.1109/TIP.2017.2655861> (дата звернення: 10.12.2024).
10. Гнатюк О. О. Технології обробки зображень в системах відеоспостереження / О. О. Гнатюк. — Київ: Наукова думка, 2018. — 330 с.
11. Баранов С. А. Системи відеоспостереження: проектування та реалізація / С. А. Баранов. — Одеса : Астрон, 2013. — 300 с.
12. Xu, L., & Zheng, X. Image Denoising Based on Non-local Means and Total Variation [Електронний ресурс] // Signal Processing. – 2019. – Vol. 159. – P. 129-136. – URL: <https://doi.org/10.1016/j.sigpro.2018.10.025> (дата звернення: 10.12.2024)
13. Кривенко О. П. Методики обробки зображень в системах відеоспостереження / О. П. Кривенко. — К. : Техніка, 2015. — 320 с.
14. Грабовський І. М. Обробка зображень в системах відеоспостереження / І. М. Грабовський. — Львів : Растр-7, 2018. — 200 с.
15. Gonzalez R. C., Woods R. E. Digital Image Processing / R. C. Gonzalez, R. E. Woods. — 4th ed. — Upper Saddle River, NJ : Pearson, 2018. — 1104 p.
16. Пташник І. І. Основи теорії обробки зображень / І. І. Пташник. — Львів : ЛНУ, 2019. — 275 с.
17. Копанівський В. В. Системи автоматизованого відеоспостереження : підруч. / В. В. Копанівський. — К. : Техніка, 2016. — 240 с.
18. Бобровський О. Г. Фільтрація зображень на основі адаптивних алгоритмів / О. Г. Бобровський, В. В. Слюсарчук // Науковий вісник НЛТУ України. — 2016. — Вип. 26. — С. 35-40.
19. Трещов В. О. Технології обробки зображень : підручник / В. О. Трещов, А. І. Шевченко. — Київ : Алерта, 2011. — 412 с.
20. Тихомиров В. П. Системи обробки зображень : підручник / В. П.

- Тихомиров. — Дніпро : ДНУ, 2014. — 500 с.
21. Гречаник А. М. Вейвлети в обробці зображень / А. М. Гречаник, С. В. Гребенюк. — Х. : Видавництво ХНУ, 2011. — 180 с.
 22. Zhou W. An Adaptive Image Denoising Algorithm Based on Wavelet Thresholding / W. Zhou, H. Chen. — 2016. — URL: <https://www.sciencedirect.com/science/article/abs/pii/S089360801530084X> (дата звернення: 10.12.2024)
 23. Гупаленко В. О. Вейвлет-аналіз: теорія та застосування / В. О. Гупаленко, О. А. Гупаленко. — К. : Наукова думка, 2010. — 350 с.
 24. Vennelakanti, S., & Lakshmanan, V. Performance analysis of wavelet-based image denoising methods. International Journal of Computer Applications, 2013, 69(1), 24–30. — DOI: 10.5120/11871-0208.
 25. Шевчук О. В. Обробка зображень у системах автоматичного контролю / О. В. Шевчук. — Київ : Видавництво КНУ, 2019. — 150 с.
 26. Олійник М. В. Вейвлети в обробці сигналів і зображень / М. В. Олійник. — Харків : ХНУ, 2016. — 280 с.
 27. Гармаш В. Метод фільтрації цифрових кольорових зображень [Електронний ресурс] / В. Гармаш, С. Пучко // Матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 27-28 квітня 2020 р. — Електрон. текст. дані. — 2020. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/8810> (дата звернення: 10.12.2024)
 28. Гармаш В.В. Метод фільтрації цифрових кольорових зображень / В. В. Гармаш, С. В. Пучко [Електронний ресурс]: XLIX Науково-технічна конференція факультету комп'ютерних систем і автоматики (2020). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/8810/7864> (дата звернення: 10.12.2024)
 29. Shevchuk V.V. Color image denoising using “chanell method” / Shevchuk V.V. Garmash V. V. // Vdecky pokrok na prelomu tysyachalety – 2011. – Р.

3 – 6. – ISBN 978-966-8736-05-6.

30. Гармаш В.В. Метод фільтрації кольорових зображень на основі моделі YCrCb/ В.В. Гармаш // Вісник Черкаського технологічного університету. Серія "Технічні науки". – 2012. – №3. – С. 69–73. – ISSN 2306-4412.
31. Барабан М. В. Метод фільтрації цифрових кольорових зображень, отриманих в умовах недостатнього рівня освітлення [Текст] / М. В. Барабан, О. М. Бевз, В. В. Гармаш // Оптико-електронні інформаційно-енергетичні технології. – 2017. – № 2. – С. 5-10.
32. Заводовський А. О., Гнатенко А. В. Аналіз алгоритмів обробки зображень для систем відеоспостереження / А. О. Заводовський, А. В. Гнатенко // Вісник Національного університету “Львівська політехніка”. — 2017. — № 858. — С. 76–82.
33. Михайлюк О. О., Романов В. І. Системи автоматизованого відеоспостереження: особливості проектування та реалізації / О. О. Михайлюк, В. І. Романов. — К. : КНУ, 2015. — 180 с.
34. The Importance of Image Filtering in Video Surveillance [Електронний ресурс] // Journal of Security Technology. – 2021. – URL: <https://www.journalofsecuritytechnology.com/image-filtering-video-surveillance/> (дата звернення: 10.12.2024)
35. The USC-SIPI Image Database. [Електронний ресурс] URL: <http://sipi.usc.edu/database/> (дата звернення: 10.12.2024)
36. International Telecommunication Union. Recommendation ITU-R BT.500-11. Methodology for the subjective assessment of the quality of television pictures [Електронний ресурс] // ITU. – 2002. – 65 с. – URL: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-11-200206-S!!PDF-E.pdf (дата звернення: 10.12.2024)
37. Moorthy A. K. A Two-Step Framework for Constructing Blind Image Quality Indices / A. K. Moorthy, A. C. Bovik // IEEE Signal Processing

- Letters. – May 2010. – Vol. 17. – №5. – P. 513 – 516. – ISSN 1070-9908.
38. Салімонович А. О., Гармаш В. В. «Метод фільтрації цифрових зображень на основі білатерального фільтру», «Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)», 2024. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20377> (дата звернення: 10.12.2024)
 39. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
 40. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.
 41. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальностей: 126 — «Інформаційні системи та технології», 151 — «Автоматизація та комп'ютерно-інтегровані технології», 174 — «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» [Електронний ресурс] / Укл. О. В. Біськало, Ю. Ю. Іванов, Р. В. Маслій. — Вінниця : ВНТУ, 2023. — 63 с. — URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/12345> (дата звернення: 10.12.2024).

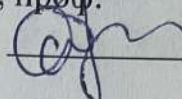
ДОДАТКИ

Додаток А (обов'язковий)
Технічне завдання на магістерську кваліфікаційну роботу

ЗАТВЕРДЖЕНО

Зав. кафедри АІТ,

д.т.н., проф.

 Олег БІСКАЛО

“ 14 ” жовтня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційну роботу

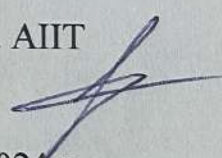
Розробка модуля фільтрації цифрових зображень для системи

автоматизації відділу відеоаналітики підприємства

08-31.МКР.005.02.000 ТЗ

Керівник роботи

к.т.н. доц. каф. АІТ

Гармаш В.В. 

«11» жовтня 2024 р.

Виконавець

ст. групи 1АКІТР-23м

Салімонович А. О. 

«11» жовтня 2024 р.

1. Назва та галузь застосування

Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства. Розроблене програмне забезпечення застосовуватиметься в системах відеоспостереження та відеоаналітики.

2. Підстава проведення для розробки.

Підставою для виконання роботи є наказ №310 по ВНТУ від 17.09. 2024 р.

3. Мета та призначення розробки.

Метою роботи є розробка модуля фільтрації цифрових зображень, який підвищить ефективність оброблення даних у системах автоматизації.

Призначенням розробки є створення програмного забезпечення, яке би реалізовувало метод фільтрації цифрових зображень.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Gonzalez R. C., Woods R. E. Digital Image Processing / R. C. Gonzalez, R. E. Woods. — 4th ed. — Upper Saddle River, NJ : Pearson, 2018. — 1104 p.

2. Добеши І. Обробка зображень в системах безпеки / І. Добеши. — Х.: ХНУРЕ, 2015. — 320 с.

3. Салімонович А. О., Гармаш В. В. «Метод фільтрації цифрових зображень на основі білатерального фільтру», «Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)», 2024. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20377>

4. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальностей: 126 — «Інформаційні системи та технології», 151 — «Автоматизація та комп'ютерно-інтегровані технології», 174 — «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» [Електронний ресурс] / Укл. О. В. Біскало, Ю. Ю. Іванов, Р. В. Маслій. — Вінниця: ВНТУ, 2023. — 63 с.

5. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В.Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.

5. Вихідні дані:

- тип цифрового зображення – півтонові та кольорові;
- мінімальна роздільна здатність зображення 128 x 128 пікселів,
- максимальна роздільна здатність 2048 x 2048 пікселів;
- рівень шуму – 5 – 70%.

6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 500 тис. грн.
- внутрішня економічна дохідність інвестицій – $\leq 0,57$,
- абсолютний економічний ефект для потенційного інвестора – не менше 1 млн. грн.
- термін окупності – не більше 3 років

7. Стадії та етапи розробки.

а) Аналіз існуючих модулів та методів фільтрації цифрових зображень до 11.10.2024 р.

б) Розробка методу фільтрації на основі білатерального фільтру до 26.10.2024 р.

в) Розробка алгоритмічного та програмного забезпечення до 15.11.2024
р.

г) Економічний розділ до 27.11.2024 р.

д) Оформлення пояснювальної записки до 01.12.2024 р.

8. Порядок контролю і приймання.

Рубіжний контроль провести до 15.11.2024 р.

Попередній захист МКР провести до 05.12.2024 р.

Захист МКР провести до 20.12.2024 р.

Додаток Б
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
МЕТОД ФІЛЬТРАЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ НА ОСНОВІ
БІЛАТЕРАЛЬНОГО ФІЛЬТРУ

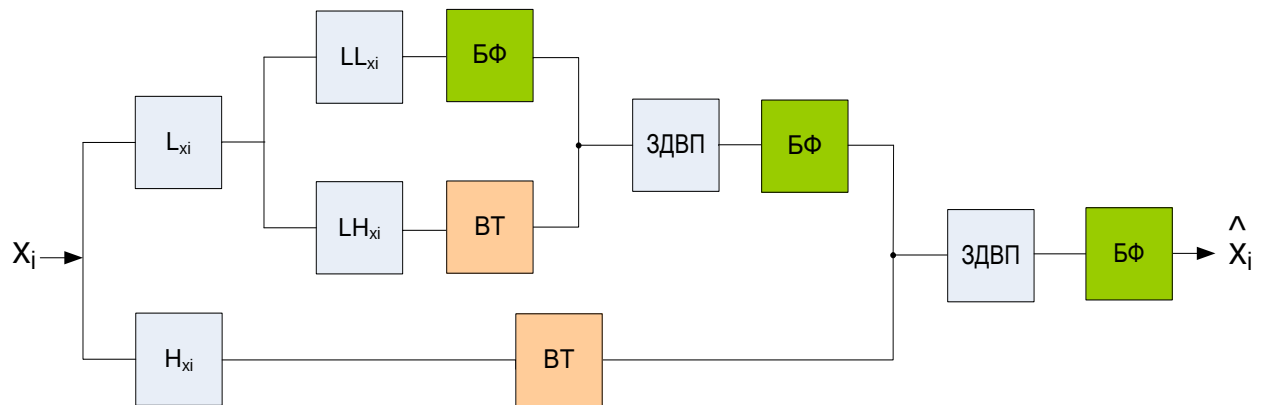


Рисунок Б1 – Процес фільтрації кратномасштабним білатеральним фільтром

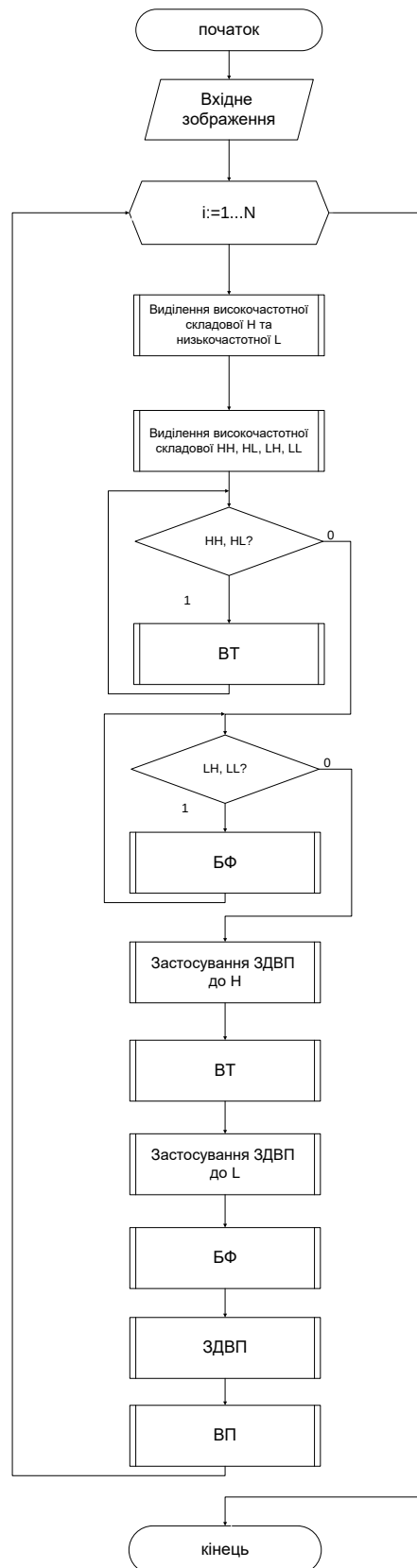


Рисунок Б2 – Метод фільтрації на основі білатерального фільтру. Схема програми.

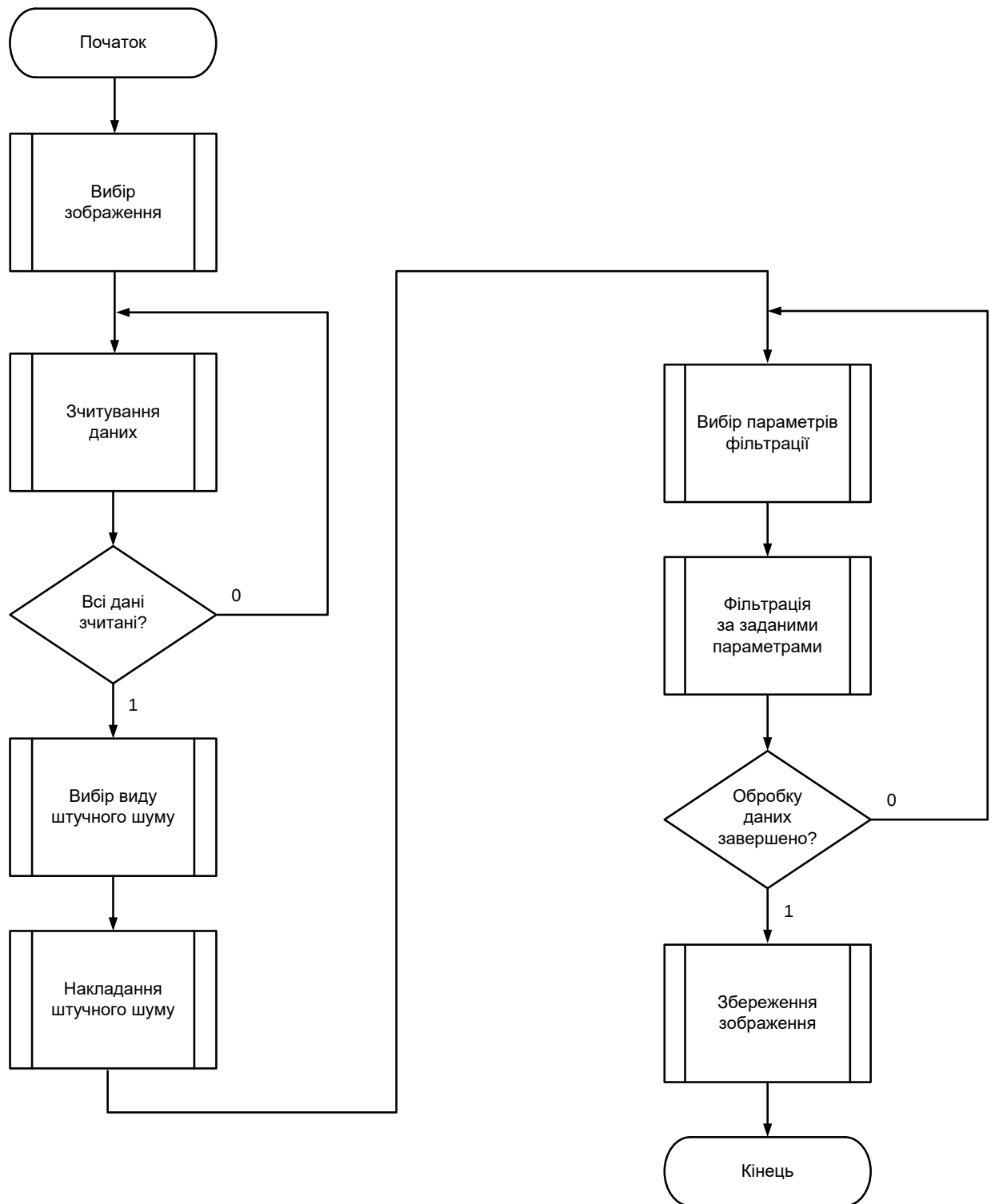


Рисунок Б.3 – Процес фільтрації цифрових зображень зі штучним шумом. Схема програми.

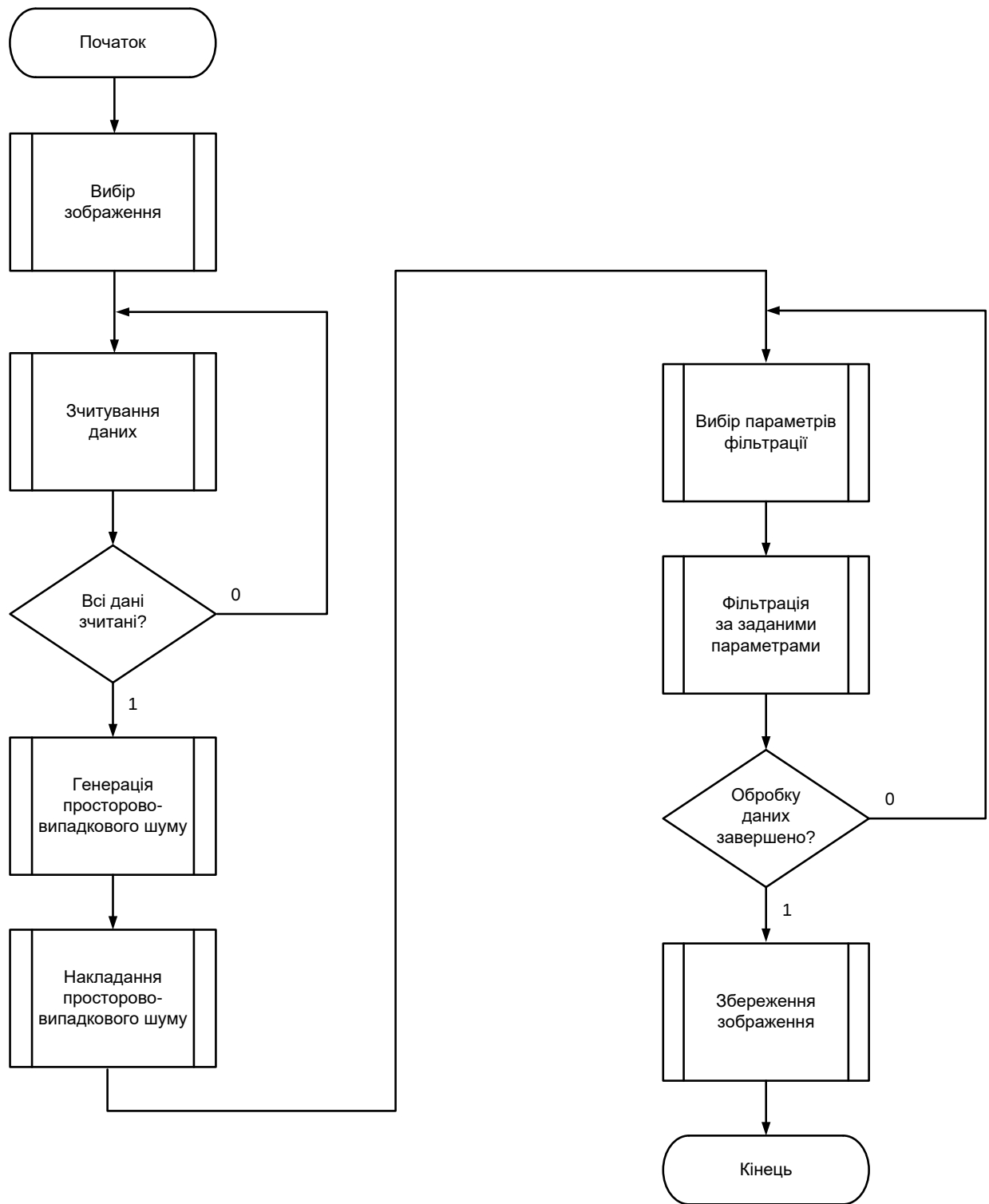


Рисунок Б.4 – Процес фільтрації цифрових зображень з просторово-випадковим шумом. Схема програми.



Рисунок Б.5 – Структура роботи підсистеми відеоспостереження з модулем фільтрації зображень

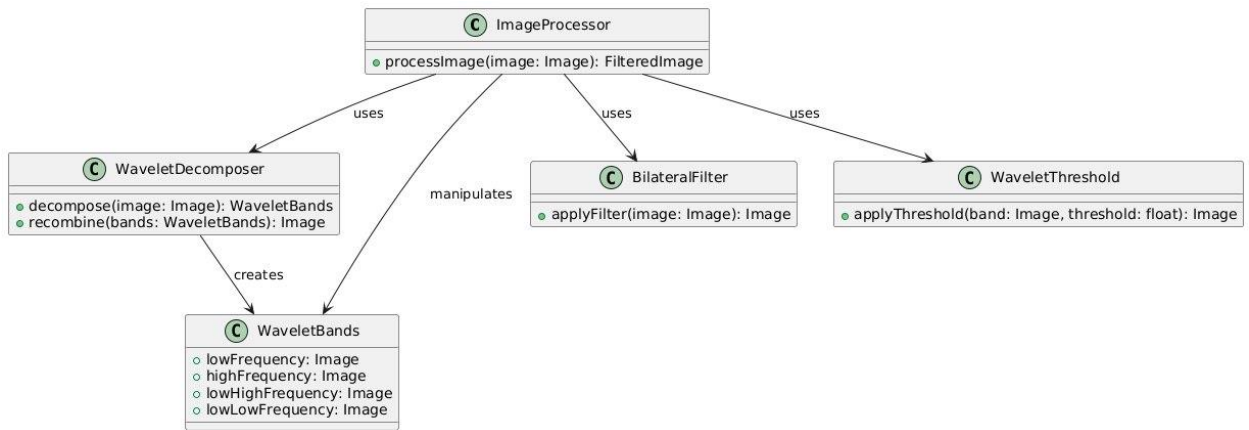


Рисунок Б.6 – Діаграма класів процесів фільтрації цифрових зображень зі штучним шумом

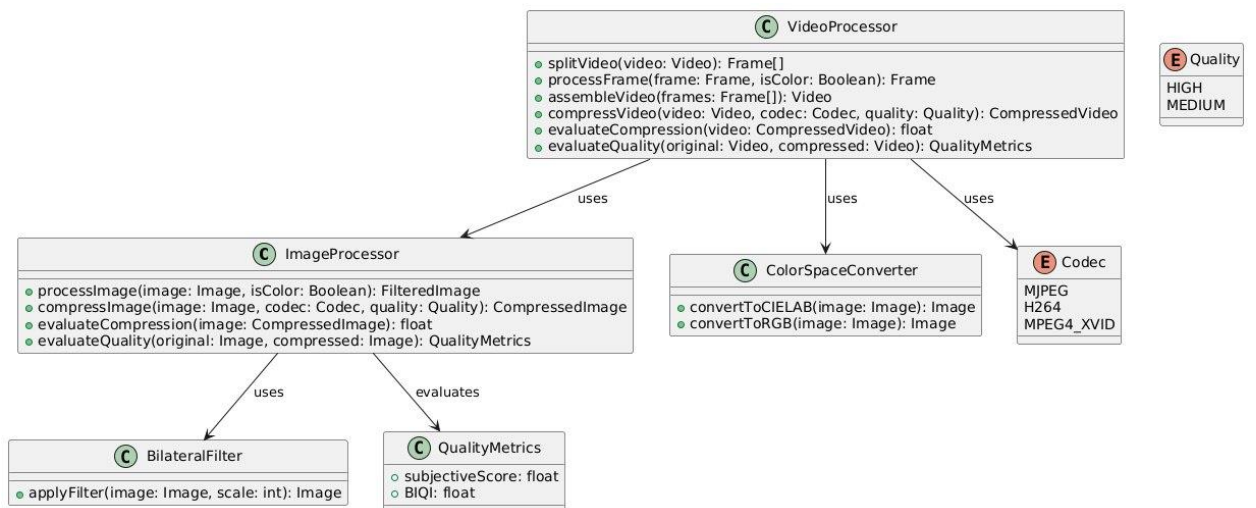


Рисунок Б.7 – Діаграма класів методики дослідження модуля фільтрації цифрових зображень



Рисунок Б.8 – Use case діаграма процесу фільтрації цифрового зображення

Додаток В (обов'язковий)

Лістинг модуля програми

```

#pragma once
#include "options.h"
#include "kodirovka.h"
#include "dekodirovka.h"
namespace video {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для Form1
    /// </summary>
    public ref class Form1 : public System::Windows::Forms::Form
    {
    public:
        Form1(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }

    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~Form1()
        {
            if (components)
            {
                delete components;
            }
        }

    private: System::Windows::Forms::MenuStrip^ menuStrip1;
    protected:
    private: System::Windows::Forms::ToolStripMenuItem^ файлToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ завантажитиToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ фотоToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ відеоToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ вихідToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ опціїToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ параметриToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ допомогаToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ допомогаToolStripMenuItem1;
    private: System::Windows::Forms::ToolStripMenuItem^ проПпрограмуToolStripMenuItem;
    private: System::Windows::Forms::Button^ button1;
    private: System::Windows::Forms::Button^ button2;
    private: System::Windows::Forms::Button^ button3;
    private: System::Windows::Forms::Button^ button4;

```

```

private: System::Windows::Forms::Button^ button5;
private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::Button^ button7;
private: System::Windows::Forms::Button^ button8;
private: System::Windows::Forms::Button^ button9;
private: System::Windows::Forms::Button^ button10;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::Button^ button11;
private: System::Windows::Forms::Button^ button12;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog2;
private: System::Windows::Forms::ToolStripMenuItem^ кодуванняToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^ декодуванняToolStripMenuItem;

private:
    /// <summary>
    /// Требуется переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Обязательный метод для поддержки конструктора - не изменяйте
    /// содержимое данного метода при помощи редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(Form1::typeid));
        this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
        this->файлToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->завантажитиToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->фотоToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->відеоToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->вихідToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->опціїToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->параметриToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->кодуванняToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->декодуванняToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->допомогаToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->допомогаToolStripMenuItem1 = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->проПрограмуToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());

```

```

this->button1 = (gcnew System::Windows::Forms::Button());
this->button2 = (gcnew System::Windows::Forms::Button());
this->button3 = (gcnew System::Windows::Forms::Button());
this->button4 = (gcnew System::Windows::Forms::Button());
this->button5 = (gcnew System::Windows::Forms::Button());
this->button6 = (gcnew System::Windows::Forms::Button());
this->button7 = (gcnew System::Windows::Forms::Button());
this->button8 = (gcnew System::Windows::Forms::Button());
this->button9 = (gcnew System::Windows::Forms::Button());
this->button10 = (gcnew System::Windows::Forms::Button());
this->textBox1 = (gcnew System::Windows::Forms::TextBox());
this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
this->button11 = (gcnew System::Windows::Forms::Button());
this->button12 = (gcnew System::Windows::Forms::Button());
this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());
this->openFileDialog2 = (gcnew System::Windows::Forms::OpenFileDialog());
this->menuStrip1->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox1))->BeginInit();

this->SuspendLayout();
//
// menuStrip1
//
this->menuStrip1->BackColor = System::Drawing::Color::Gray;
this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(3) { this->файлToolStripMenuItem,
this->опціїToolStripMenuItem, this->допомогаToolStripMenuItem});
this->menuStrip1->Location = System::Drawing::Point(0, 0);
this->menuStrip1->Name = L"menuStrip1";
this->menuStrip1->Size = System::Drawing::Size(982, 24);
this->menuStrip1->TabIndex = 0;
this->menuStrip1->Text = L"menuStrip1";
//
// файлToolStripMenuItem
//
this->файлToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^ >(2) { this->завантажитиToolStripMenuItem,
this->вихідToolStripMenuItem});
this->файлToolStripMenuItem->ForeColor =
System::Drawing::SystemColors::ButtonHighlight;
this->файлToolStripMenuItem->Name = L"файлToolStripMenuItem";
this->файлToolStripMenuItem->Size = System::Drawing::Size(48, 20);
this->файлToolStripMenuItem->Text = L"Файл";
//
// завантажитиToolStripMenuItem
//
this->завантажитиToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^ >(2) { this->фотоToolStripMenuItem,
this->відеоToolStripMenuItem});
this->завантажитиToolStripMenuItem->Name =
L"завантажитиToolStripMenuItem";
this->завантажитиToolStripMenuItem->Size = System::Drawing::Size(152,
22);

this->завантажитиToolStripMenuItem->Text = L"Завантажити";
//
// фотоToolStripMenuItem
//
this->фотоToolStripMenuItem->Name = L"фотоToolStripMenuItem";
this->фотоToolStripMenuItem->Size = System::Drawing::Size(103, 22);
this->фотоToolStripMenuItem->Text = L"Фото";

```

```

//
// відеоToolStripMenuItem
//
this->відеоToolStripMenuItem->Name = L"відеоToolStripMenuItem";
this->відеоToolStripMenuItem->Size = System::Drawing::Size(103, 22);
this->відеоToolStripMenuItem->Text = L"Відео";
//
// вихідToolStripMenuItem
//
this->вихідToolStripMenuItem->Name = L"вихідToolStripMenuItem";
this->вихідToolStripMenuItem->Size = System::Drawing::Size(152, 22);
this->вихідToolStripMenuItem->Text = L"Вихід";
//
// опціїToolStripMenuItem
//
this->опціїToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^ >(3) { this->параметриToolStripMenuItem,
this->кодуванняToolStripMenuItem, this-
>декодуванняToolStripMenuItem});
this->опціїToolStripMenuItem->ForeColor =
System::Drawing::SystemColors::ButtonHighlight;
this->опціїToolStripMenuItem->Name = L"опціїToolStripMenuItem";
this->опціїToolStripMenuItem->Size = System::Drawing::Size(48, 20);
this->опціїToolStripMenuItem->Text = L"Опції";
//
// параметриToolStripMenuItem
//
this->параметриToolStripMenuItem->Name =
L"параметриToolStripMenuItem";
this->параметриToolStripMenuItem->Size = System::Drawing::Size(152, 22);
this->параметриToolStripMenuItem->Text = L"Фільтри";
this->параметриToolStripMenuItem->Click += gcnew
System::EventHandler(this, &Form1::параметриToolStripMenuItem_Click);
//
// кодуванняToolStripMenuItem
//
this->кодуванняToolStripMenuItem->Name =
L"кодуванняToolStripMenuItem";
this->кодуванняToolStripMenuItem->Size = System::Drawing::Size(152, 22);
this->кодуванняToolStripMenuItem->Text = L"Кодування";
this->кодуванняToolStripMenuItem->Click += gcnew
System::EventHandler(this, &Form1::кодуванняToolStripMenuItem_Click);
//
// декодуванняToolStripMenuItem
//
this->декодуванняToolStripMenuItem->Name =
L"декодуванняToolStripMenuItem";
this->декодуванняToolStripMenuItem->Size = System::Drawing::Size(152,
22);
this->декодуванняToolStripMenuItem->Text = L"Декодування";
this->декодуванняToolStripMenuItem->Click += gcnew
System::EventHandler(this, &Form1::декодуванняToolStripMenuItem_Click);
//
// допомогаToolStripMenuItem
//
this->допомогаToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^ >(2) { this->допомогаToolStripMenuItem1,
this->проПрограмуToolStripMenuItem});
this->допомогаToolStripMenuItem->ForeColor =
System::Drawing::SystemColors::ButtonHighlight;

```

```

this->допомогаToolStripMenuItem->Name = L"допомогаToolStripMenuItem";
this->допомогаToolStripMenuItem->Size = System::Drawing::Size(75, 20);
this->допомогаToolStripMenuItem->Text = L"Допомога";
//
// допомогаToolStripMenuItem
//
this->допомогаToolStripMenuItem1->Name =
L"допомогаToolStripMenuItem1";
this->допомогаToolStripMenuItem1->Size = System::Drawing::Size(154, 22);
this->допомогаToolStripMenuItem1->Text = L"Допомога";
//
// проПрограмуToolStripMenuItem
//
this->проПрограмуToolStripMenuItem->Name =
L"проПрограмуToolStripMenuItem";
this->проПрограмуToolStripMenuItem->Size = System::Drawing::Size(154,
22);
this->проПрограмуToolStripMenuItem->Text = L"Про програму";
//
// button1
//
this->button1->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button1.BackgroundImage")));
this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button1->ForeColor =
System::Drawing::SystemColors::ActiveCaptionText;
this->button1->Location = System::Drawing::Point(12, 43);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(205, 50);
this->button1->TabIndex = 1;
this->button1->Text = L" Додати зображення";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this,
&Form1::button1_Click);
//
// button2
//
this->button2->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button2.BackgroundImage")));
this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button2->ForeColor =
System::Drawing::SystemColors::ActiveCaptionText;
this->button2->Location = System::Drawing::Point(12, 110);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(205, 50);
this->button2->TabIndex = 2;
this->button2->Text = L" Додати відеопослідовність";
this->button2->UseVisualStyleBackColor = true;
//
// button3
//
this->button3->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button3.BackgroundImage")));
this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button3->ForeColor = System::Drawing::Color::OrangeRed;
this->button3->Location = System::Drawing::Point(12, 176);
this->button3->Name = L"button3";
this->button3->Size = System::Drawing::Size(205, 50);
this->button3->TabIndex = 3;
this->button3->Text = L"Видалити";

```

```

        this->button3->UseVisualStyleBackColor = true;
        //
        // button4
        //
        this->button4->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button4.BackgroundImage")));
        this->button4->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button4->ForeColor =
System::Drawing::SystemColors::ActiveCaptionText;
        this->button4->Location = System::Drawing::Point(12, 243);
        this->button4->Name = L"button4";
        this->button4->Size = System::Drawing::Size(205, 50);
        this->button4->TabIndex = 4;
        this->button4->Text = L"Видалити всі";
        this->button4->UseVisualStyleBackColor = true;
        //
        // button5
        //
        this->button5->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button5.BackgroundImage")));
        this->button5->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button5->ForeColor = System::Drawing::SystemColors::HighlightText;
        this->button5->Location = System::Drawing::Point(12, 347);
        this->button5->Name = L"button5";
        this->button5->Size = System::Drawing::Size(205, 50);
        this->button5->TabIndex = 5;
        this->button5->Text = L"Фільтри";
        this->button5->UseVisualStyleBackColor = true;
        this->button5->Click += gcnew System::EventHandler(this,
&Form1::button5_Click);
        //
        // button6
        //
        this->button6->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom |
System::Windows::Forms::AnchorStyles::Left));
        this->button6->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button6.BackgroundImage")));
        this->button6->FlatAppearance->MouseDownBackColor =
System::Drawing::SystemColors::WindowFrame;
        this->button6->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button6->ForeColor = System::Drawing::Color::Green;
        this->button6->Location = System::Drawing::Point(12, 571);
        this->button6->Name = L"button6";
        this->button6->Size = System::Drawing::Size(205, 50);
        this->button6->TabIndex = 6;
        this->button6->Text = L"Розпочати";
        this->button6->UseVisualStyleBackColor = true;
        //
        // button7
        //
        this->button7->Anchor = System::Windows::Forms::AnchorStyles::Bottom;
        this->button7->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button7.BackgroundImage")));
        this->button7->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button7->Location = System::Drawing::Point(440, 571);
        this->button7->Name = L"button7";
        this->button7->Size = System::Drawing::Size(50, 50);
        this->button7->TabIndex = 7;
        this->button7->UseVisualStyleBackColor = true;

```

```

//
// button8
//
this->button8->Anchor = System::Windows::Forms::AnchorStyles::Bottom;
this->button8->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button8.BackgroundImage"))));
this->button8->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button8->Location = System::Drawing::Point(496, 571);
this->button8->Name = L"button8";
this->button8->Size = System::Drawing::Size(50, 50);
this->button8->TabIndex = 8;
this->button8->UseVisualStyleBackColor = true;
//
// button9
//
this->button9->Anchor = System::Windows::Forms::AnchorStyles::Bottom;
this->button9->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button9.BackgroundImage"))));
this->button9->FlatAppearance->MouseDownBackColor =
System::Drawing::Color::Silver;
this->button9->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button9->Location = System::Drawing::Point(658, 571);
this->button9->Name = L"button9";
this->button9->Size = System::Drawing::Size(50, 50);
this->button9->TabIndex = 9;
this->button9->UseMnemonic = false;
this->button9->UseVisualStyleBackColor = false;
//
// button10
//
this->button10->Anchor = System::Windows::Forms::AnchorStyles::Bottom;
this->button10->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button10.BackgroundImage"))));
this->button10->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
this->button10->Location = System::Drawing::Point(714, 571);
this->button10->Name = L"button10";
this->button10->Size = System::Drawing::Size(50, 50);
this->button10->TabIndex = 10;
this->button10->UseVisualStyleBackColor = true;
//
// textBox1
//
this->textBox1->Anchor = System::Windows::Forms::AnchorStyles::Bottom;
this->textBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;
this->textBox1->Location = System::Drawing::Point(567, 592);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(62, 13);
this->textBox1->TabIndex = 11;
this->textBox1->Text = L"1/1";
this->textBox1->TextAlign =
System::Windows::Forms::HorizontalAlignment::Center;
//
// pictureBox1
//
this->pictureBox1->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top |
System::Windows::Forms::AnchorStyles::Bottom)
| System::Windows::Forms::AnchorStyles::Left)
| System::Windows::Forms::AnchorStyles::Right));

```

```

        this->pictureBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Center;
        this->pictureBox1->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->pictureBox1->Location = System::Drawing::Point(228, 43);
        this->pictureBox1->Margin = System::Windows::Forms::Padding(5);
        this->pictureBox1->MinimumSize = System::Drawing::Size(360, 482);
        this->pictureBox1->Name = L"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(740, 520);
        this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::StretchImage;
        this->pictureBox1->TabIndex = 12;
        this->pictureBox1->TabStop = false;
        //
        // button11
        //
        this->button11->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button11.BackgroundImage")));
        this->button11->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button11->ForeColor = System::Drawing::Color::DarkBlue;
        this->button11->Location = System::Drawing::Point(12, 457);
        this->button11->Name = L"button11";
        this->button11->Size = System::Drawing::Size(205, 50);
        this->button11->TabIndex = 16;
        this->button11->Text = L"Зберегти зміни";
        this->button11->UseVisualStyleBackColor = true;
        //
        // button12
        //
        this->button12->BackgroundImage = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"button12.BackgroundImage")));
        this->button12->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
        this->button12->ForeColor = System::Drawing::Color::Red;
        this->button12->Location = System::Drawing::Point(12, 513);
        this->button12->Name = L"button12";
        this->button12->Size = System::Drawing::Size(205, 50);
        this->button12->TabIndex = 17;
        this->button12->Text = L"Відмінити";
        this->button12->UseVisualStyleBackColor = true;
        //
        // Form1
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->AutoScroll = true;
        this->BackColor = System::Drawing::SystemColors::Window;
        this->ClientSize = System::Drawing::Size(982, 633);
        this->Controls->Add(this->button12);
        this->Controls->Add(this->button11);
        this->Controls->Add(this->pictureBox1);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->button10);
        this->Controls->Add(this->button9);
        this->Controls->Add(this->button8);
        this->Controls->Add(this->button7);
        this->Controls->Add(this->button6);
        this->Controls->Add(this->button5);
        this->Controls->Add(this->button4);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);

```

```

        this->Controls->Add(this->button1);
        this->Controls->Add(this->menuStrip1);
        this->MainMenuStrip = this->menuStrip1;
        this->MinimumSize = System::Drawing::Size(982, 672);
        this->Name = L "Form1";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L "Бакалаврська Зінченко";
        this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>pictureBox1))->EndInit();

        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) {
        options^ Form2=gcnew options();
        Form2->ShowDialog();
    }
    private: System::Void параметриToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
        options^ Form2=gcnew options();
        Form2->ShowDialog();
    }
    private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
        openFileDialog1->ShowDialog();
        if (openFileDialog1->FileName!="")
            pictureBox1->Load(openFileDialog1->FileName);
    }

    private: System::Void кодунанняToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
        kodirovka^ kod = gcnew kodirovka();
        kod->ShowDialog();
    }
    private: System::Void декодуванняToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        dekodirovka^ dek= gcnew dekodirovka();
        dek->ShowDialog();
    }
    private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    }
};
}

```

Додаток Г
(обов'язковий)

ПРОТОКОЛ
ПЕРЕВІРКИ ДИПЛОМНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Розробка модуля фільтрації цифрових зображень для системи автоматизації відділу відеоаналітики підприємства»

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: кафедра АІТ, ФІТА, ІАКІТР-23м
(кафедра, факультет, навчальна група)

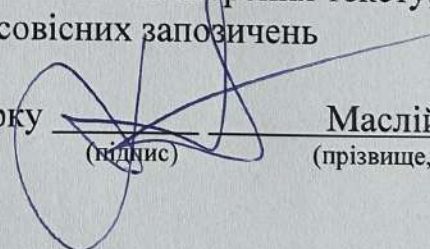
Науковий керівник: Гармаш В.В., доцент каф. АІТ
(прізвище, ініціали, посада)

Показники звіту подібності Turnitin

Оригінальність 81 % Схожість 19 %

Аналіз звіту подібності (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень

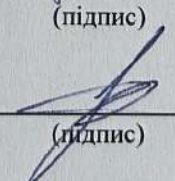
Особа, відповідальна за перевірку  Маслій Р. В.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи

Автор

 Салімонович А.О.
(підпис) (прізвище, ініціали)

Керівник роботи

 Гармаш В.В.
(підпис) (прізвище, ініціали)